

TITLE1**AH-B113A-BE
 TITLE2**VAX/VMS V4.0
 FICHENUMBER**1

[illegible]

```
AAAAAA  CCCCCCCC  CCCCCCCC
AAAAAA  CCCCCCCC  CCCCCCCC
AA      AA  CC      CC
AA      AA  CC      CC
AA      AA  CC      CC
AA      AA  CC      CC
AA      AA  CC      CC
AA      AA  CC      CC
AAAAAAAA  CC      CC
AAAAAAAA  CC      CC
AA      AA  CC      CC
AA      AA  CC      CC
AA      AA  CC      CC
AA      AA  CCCCCCCC  CCCCCCCC
AA      AA  CCCCCCCC  CCCCCCCC
```

....
....
....
....

```
LL      IIIIII  SSSSSSSS
LL      IIIIII  SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLL  IIIIII  SSSSSSSS
LLLLLLLL  IIIIII  SSSSSSSS
```

```
0001 0 MODULE
0002 0 ACC (IDENT = 'V04-000', MAIN = ACC) =
0003 1 BEGIN
0004 1
0005 1
0006 1 *****
0007 1 *
0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0010 1 * ALL RIGHTS RESERVED.
0011 1 *
0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0017 1 * TRANSFERRED.
0018 1 *
0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0021 1 * CORPORATION.
0022 1 *
0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0025 1 *
0026 1 *****
0027 1
0028 1
0029 1 ++
0030 1 FACILITY: ACC, Account file dumper
0031 1
0032 1 ABSTRACT:
0033 1
0034 1 This utility dumps the system accounting file in a readable, labelled
0035 1 format, and allows searching for particular entries. Also, two
0036 1 binary output files may be created containing the selected and
0037 1 unselected records respectively.
0038 1
0039 1 ENVIRONMENT:
0040 1
0041 1 VAX/VMS operating system. unprivileged user mode.
0042 1
0043 1 AUTHOR: Greg Robert and Steve Forgey, January 1982
0044 1
0045 1 Modified by:
0046 1
0047 1 V03-021 DAS0010 David Solomon 16-Jul-1984
0048 1 Translate UIC from binary to alphanumeric only for /SUMMARY=UIC.
0049 1 Do translation for /FULL in FORMAT\SHOW_FULL only if we need to
0050 1 display it.
0051 1
0052 1 V03-020 DAS0009 David Solomon 09-Jul-1984
0053 1 Fix bug in string comparison of '-' for /UIC (SPR 11-68680).
0054 1
0055 1 V03-019 DAS0008 David Solomon 28-Jun-1984
0056 1 Fix order of key table so that summary by UIC, elapsed time,
0057 1 priority, volumes, pages printed, and symbiont gets and qios
```

operates on the correct summary key. Close file before returning to LIB\$FILE_SCAN. SPR 11-57832: 1) increase size of elapsed time and CPU time accumulator from 3 to 4 longwords. 2) signal errors during PROCESS_FILE instead of ignoring them and stopping processing of the input file. SPR 11-CS00112: fix UIC exclusion.

V03-018 DAS0007 David Solomon 27-Jun-1984
Only translate UIC if /SUMMARY=UIC instead of just /SUMMARY.

V03-017 DAS0006 David Solomon 12-Apr-1984
Fix bug parsing image filespecs (due to image filespecs now being unconcealed). Use \$FILESCAN to parse it now.

V03-016 DAS0005 David Solomon 28-Feb-1984
Add context longword to call to LIB\$FILE_SCAN to make stickyness work correctly with search lists.

V03-015 DAS0004 David Solomon 23-Jan-1984
Extend length of queue names to 31 and job names to 39 bytes. Don't perform conversion to alphanumeric UICs unless doing summary or full output. Fix bug where selection of a zero group or member UIC number was resulting in wildcard selection for that field.

V03-014 DAS0003 David Solomon 12-Jan-1984
Get ACCDEF.REQ from SRC\$, not MSRC\$.

V03-013 DAS0002 David Solomon 02-Jan-1984
Use unique symbol table index for summation table (was using 7, which was the index for the /IDENTIFICATION symbol table). If no files are selected, exit immediately (don't incorrectly call SORT routines, try and close output files, etc.). Improve recovery from bogus records in accounting file. Set key start position in SORT_TABLE correctly in PARSE_KEYS.

V03-012 LMP0140 L. Mark Pilant, 19-Aug-1983 11:09
Add support for alphanumeric UICs.

V03-011 DAS0001 David Solomon 25-Jul-1983
Now that the Job Controller fills in ACR\$Q_BEGTIME, correctly compute the elapsed time as end time minus begin time.

V03-010 SPF0205 Steve Forgey 29-Jan-1983
SPR #51632
Fix UIC exclusion bug.

V03-009 SPF0204 Steve Forgey 29-Jan-1983
SPR #51768
Properly handle UICs with more than 3 digits per field.

V03-008 SPF0203 Steve Forgey 29-Jan-1983
SPR #51083
Expand "days" field in time values in summary reports from 2 to 4 digits.

V03-007 SPF0202 Steve Forgey 29-Jan-1983
SPR #50610

ACC
V04-000

B 4
15-Sep-1984 23:36:11
14-Sep-1984 11:51:58

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[ACC.SRC]ACC.332;1

Page 3
(1)

```

115 0115 1 | Fix input file wilddarding problem.
116 0116 1 |
117 0117 1 | V03-006 SPF0201 Steve Forgey 29-Jan-1983
118 0118 1 | SPR #48396
119 0119 1 | Eliminate erroneous data in summary report.
120 0120 1 |
121 0121 1 | V03-005 SPF0113 Steve Forgey 19-Apr-1982
122 0122 1 | Don't use RETURN_IF_ERROR macro where the variable STATUS
123 0123 1 | is already defined.
124 0124 1 |
125 0125 1 | V03-004 SPF0112 Steve Forgey 10-Apr-1982
126 0126 1 | Format pre-Version 3 queue names as counted ASCII strings.
127 0127 1 |
128 0128 1 | V03-003 SPF0105 Steve Forgey 27-Mar-1982
129 0129 1 | Get input file defaults from previous input file specs.
130 0130 1 |
131 0131 1 | V03-002 SPF0104 Steve Forgey 27-Mar-1982
132 0132 1 | Specify correct max record size to SOR$INIT_SORT.
133 0133 1 |
134 0134 1 | V03-001 SPF0102 Steve Forgey 27-Mar-1982
135 0135 1 | Don't add null value to selection list when parsing the
136 0136 1 | exclusion qualifier, "-.".
137 0137 1 |
138 0138 1 | V02-002 SPF0081 Steve Forgey Feb-06-1982
139 0139 1 | Fix value errors.
140 0140 1 |
141 0141 1 | V02-001 SPF0070 Steve Forgey Jan-23-1982
142 0142 1 | Accept <> as directory delimiters in image file spec as well
143 0143 1 | as []. Also improve SORT performance.
144 0144 1 |
145 0145 1 | --
146 0146 1 |
147 0147 1 | ----->
148 0148 1 |
149 0149 1 | INCLUDE FILES
150 0150 1 | ----->
151 0151 1 |
152 0152 1 |
153 0153 1 | REQUIRE 'SRC$:ACCDEF'; ! Common ACC definitions
```

TABLE OF CONTENTS

FORWARD ROUTINE

ACC,	Top level routine
ACC_CONTROL,	Main control loop
PROCESS_FILE,	Processes one input file
PROCESS_RECORD,	Analyze input record
PROCESS_V2_RECORD,	Analyze V2 record
PROCESS_V3_RECORD,	Analyze V3 record
RECORD_SELECTED,	Determine output disposition
PARSE_COMMAND,	Analyze command line
PARSE_DATES,	Handles /DATE, /BEFORE, /SINCE
PARSE_KEYS;	Handles /REPORT, /SORT, /SUMMARY

174
175
176
177
178
179
180
181
182
183
184
185
186
187

1010 1
1011 1
1012 1
1013 1
1014 1
1015 1
1016 1
1017 1
1018 1
1019 1
1020 1
1021 1
1022 1
1023 1

EXTERNAL STORAGE DEFINITIONS

EXTERNAL

INPUT_NAM: BBLOCK [],
INPUT_FAB: BBLOCK [],
INPUT_RAB: BBLOCK [],
OUTPUT_FAB: BBLOCK [],
OUTPUT_RAB: BBLOCK [],
REJECTED_FAB: BBLOCK [],
REJECTED_RAB: BBLOCK [];

GLOBAL STORAGE DEFINITIONS

GLOBAL

KEY_BUF: BBLOCK [1024],

! Key/record buffer

HOLD_BIOCNT,

HOLD_DIOCNT,

HOLD_ELAPSED: VECTOR [2],

HOLD_EXECUTION,

HOLD_PAGEFLTS,

HOLD_GETCNT,

HOLD_QIOCNT,

HOLD_PAGCNT,

HOLD_PGFLPEAK,

HOLD_PAGEREADS,

HOLD_WSPEAK,

HOLD_CPUIM: VECTOR [2],

HOLD_VOLUMES,

CUSTOMER,

RECORD_LENGTH,

RECORD_SUBTYPE,

RECORD_TYPE,

VERSION,

GROUP,

MEMBER,

CONVERTED_UIC,

QUAD_CPU_TIME: VECTOR [2],

ELAPSED_TIME: VECTOR [2],

WAIT_TIME: VECTOR [2],

ACCOUNT_DESC: VECTOR [2],

ADDRESS_DESC: VECTOR [2],

BUFFERED_IO_DESC: VECTOR [2],

CPU_TIME_DESC: VECTOR [2],

DIRECT_IO_DESC: VECTOR [2],

ENTRY_DESC: VECTOR [2],

ELAPSED_DESC: VECTOR [2],

EXECUTION_DESC: VECTOR [2],

FILE_DESC: VECTOR [2],

FINISH_TIME_DESC: VECTOR [2],

IDENT_DESC: VECTOR [2],

IMAGE_FILE_DESC: VECTOR [2],

IMAGE_DESC: VECTOR [2],

JOB_DESC: VECTOR [2],

NODE_DESC: VECTOR [2],

OWNER_DESC: VECTOR [2],

PAGE_FAULTS_DESC: VECTOR [2],

PAGE_FILE_DESC: VECTOR [2],

PAGE_READS_DESC: VECTOR [2],

! Customer type (V3 format only)

! Record length

! Record subtype (V3 format only)

! Record type

! Record format version

! UIC group

! UIC member

! Converted UIC

! CPU time (quadword format)

! Calculated elapsed time

! Calculated queue wait time

! Describes account name

! Describes remote node address

! Describes buffered I/O count

! Describes cpu time

! Describes direct I/O count

! Describes job id number

! Describes elapsed time

! Describes image execution count

! Describes file name

! Describes finish time

! Describes process id (IDENT)

! Describes image file specification

! Describes image name

! Describes job name

! Describes remote node name

! Describes owner IDENT

! Describes page fault count

! Describes page file peak usage

! Describes page read count


```
246 1081 1 PRIORITY_DESC: VECTOR [2], ! Describes process base priority
247 1082 1 PRIVILEGE_DESC: VECTOR [2], ! Describes process privilege mask
248 1083 1 PROCESS_DESC: VECTOR [2], ! Describes process type
249 1084 1 QUEUE_DESC: VECTOR [2], ! Describes queue name
250 1085 1 QUEUE_TIME_DESC: VECTOR [2], ! Describes queue time
251 1086 1 REMOTE_ID_DESC: VECTOR [2], ! Describes remote id
252 1087 1 START_TIME_DESC: VECTOR [2], ! Describes start time
253 1088 1 STATUS_DESC: VECTOR [2], ! Describes final exit status code
254 1089 1 STATUS_TEXT_DESC: VECTOR [2], ! Describes final exit status text
255 1090 1 SUBTYPE_DESC: VECTOR [2], ! Describes record subtype
256 1091 1 SYM_GET_DESC: VECTOR [2], ! Describes symbiont get count
257 1092 1 SYM_QIO_DESC: VECTOR [2], ! Describes symbiont qio count
258 1093 1 SYM_PAGE_DESC: VECTOR [2], ! Describes symbiont page count
259 1094 1 TERMINAL_DESC: VECTOR [2], ! Describes terminal name
260 1095 1 TYPE_DESC: VECTOR [2], ! Describes record type
261 1096 1 TYPE_TEXT_DESC: VECTOR [2], ! Describes full record type text
262 1097 1 UIC_BINARY_DESC: VECTOR [2], ! Describes process UIC (binary)
263 1098 1 UIC_GROUP_DESC: VECTOR [2], ! Describes process UIC group (binary)
264 1099 1 UIC_MEMBER_DESC: VECTOR [2], ! Describes process UIC member (binary)
265 1100 1 UIC_DESC: VECTOR [2], ! Describes process UIC (ascii)
266 1101 1 USER_DATA_DESC: VECTOR [2], ! Describes user data
267 1102 1 USER_DESC: VECTOR [2], ! Describes user name
268 1103 1 VOLUMES_DESC: VECTOR [2], ! Describes volume mount count
269 1104 1 WORKING_SET_DESC: VECTOR [2], ! Describes working set peak
270 1105 1
271 1106 1
272 1107 1 ! When selection qualifiers are specified these switches indicate
273 1108 1 ! whether the qualifier "selects" or "rejects" records. If true
274 1109 1 ! then members of the list are selected, if false then rejected.
275 1110 1 ! If the qualifier is not specified then the switch is never interrogated.
276 1111 1
277 1112 1
278 1113 1 ACCOUNT_SWITCH,
279 1114 1 ADDRESS_SWITCH,
280 1115 1 ENTRY_SWITCH,
281 1116 1 IDENT_SWITCH,
282 1117 1 IMAGE_SWITCH,
283 1118 1 JOB_SWITCH,
284 1119 1 NODE_SWITCH,
285 1120 1 OWNER_SWITCH,
286 1121 1 PRIORITY_SWITCH,
287 1122 1 PROCESS_SWITCH,
288 1123 1 QUEUE_SWITCH,
289 1124 1 REMOTE_ID_SWITCH,
290 1125 1 STATUS_SWITCH,
291 1126 1 TERMINAL_SWITCH,
292 1127 1 TYPE_SWITCH,
293 1128 1 UIC_SWITCH,
294 1129 1 UIC_GROUP_SWITCH,
295 1130 1 UIC_MEMBER_SWITCH,
296 1131 1 USER_SWITCH,
297 1132 1
298 1133 1 SELECTED, ! Count of selected records - this file
299 1134 1 TOTAL_SELECTED, ! Count of selected records - all files
300 1135 1 TOTAL_REJECTED, ! Count of rejected records - all files
301 1136 1
302 1137 1 SORT_KEY_SIZE, ! Total length of sort keys
```

```
303 1138 1
304 1139 1 DATE_HEADER,          ! Holds pointer to descriptor
305 1140 1
306 1141 1 STATUS_TEXT_BUFFER: BBLOCK [80],
307 1142 1
308 1143 1
309 1144 1 ! STORAGE FOR QUALIFIER VALUES AND COUNTS --
310 1145 1
311 1146 1 UIC_BUF:          VECTOR [67, BYTE],      ! process UIC (ascii)
312 1147 1 A_DATE_BUF:    VECTOR [23, BYTE],      ! DD-MMM-YYYY HH:MM:SS.CC
313 1148 1 A_DATE_DESC:   VECTOR [2]              ! Describes date in ascii
314 1149 1 INITIAL      (23, A_DATE_BUF),
315 1150 1
316 1151 1 NUMERIC_DATE: VECTOR [7, WORD],        ! Holds date in numeric format
317 1152 1                                     ! Year, month, hour, day, minute,
318 1153 1                                     ! second, hundredths -- binary words
319 1154 1
320 1155 1 S_DATE_BUF:    VECTOR [10, BYTE],      ! YYYY MM DD
321 1156 1 INITIAL      (BYTE(' ', ' ')),
322 1157 1 S_DATE_DESC:   VECTOR [2]              ! Describes date in ascii (sort format)
323 1158 1 INITIAL      (10, S_DATE_BUF),
324 1159 1
325 1160 1 YEAR_DESC:    VECTOR [2]              ! Ascii year
326 1161 1 INITIAL      (4, S_DATE_BUF + 0),
327 1162 1
328 1163 1 MONTH_DESC:   VECTOR [2]              ! Ascii number string from 01 - 12
329 1164 1 INITIAL      (2, S_DATE_BUF + 5),
330 1165 1
331 1166 1 DAY_DESC:     VECTOR [2]              ! Ascii number string from 01 - 31
332 1167 1 INITIAL      (2, S_DATE_BUF + 8),
333 1168 1
334 1169 1 WEEKDAY,      ! Day of week
335 1170 1 WEEKDAY_DESC: VECTOR [2]              ! Ascii number string from 0 - 6
336 1171 1 INITIAL      (1, WEEKDAY),
337 1172 1
338 1173 1 HOUR_DESC:    VECTOR [2]              ! Ascii number string from 00 - 23
339 1174 1 INITIAL      (2, A_DATE_BUF + 12),
340 1175 1
341 1176 1 MESSAGE_DESC: VECTOR [2],              ! Describes user message text
342 1177 1
343 1178 1 REPORT_HDR1 FAO:BBLOCK [DSC$K_D_BLN]    ! FAO for /REPORT header 1
344 1179 1 PRESET([DSC$B_CLASS] = DSC$R_CLASS_D),
345 1180 1 REPORT_HDR2 FAO:BBLOCK [DSC$K_D_BLN]    ! FAO for /REPORT header 2
346 1181 1 PRESET([DSC$B_CLASS] = DSC$R_CLASS_D),
347 1182 1 REPORT_HDR3 FAO:BBLOCK [DSC$K_D_BLN]    ! FAO for /REPORT header 3
348 1183 1 PRESET([DSC$B_CLASS] = DSC$R_CLASS_D),
349 1184 1 REPORT_DET FAO:BBLOCK [DSC$K_D_BLN]      ! FAO for /REPORT detail line
350 1185 1 PRESET([DSC$B_CLASS] = DSC$R_CLASS_D),
351 1186 1 REPORT_VALUE:   VECTOR [MAX_REPORT],      ! Holds /REPORT values
352 1187 1 REPORT_ITEMS,   ! Holds number of report items
353 1188 1 BUCKET_SIZE,    ! Size of summation bucket in bytes
354 1189 1
355 1190 1 SUM_KEY FAO:    BBLOCK [DSC$K_D_BLN]    ! FAO for /SUMMARY keys
356 1191 1 PRESET([DSC$B_CLASS] = DSC$R_CLASS_D),
357 1192 1 SUM_KEY_VALUE:   VECTOR [MAX_SUM],        ! Argument table for key FAO
358 1193 1 DATE_DESC_ADDR, ! Address of selected date desc
359 1194 1
```

```

: 360      1195 1 BEFORE_DATE:      ! Selection cutoff date
: 361      1196 1      BLOCKVECTOR [2, LONG, BYTE]      ! -allocate one quadword
: 362      1197 1      INITIAL      ! -initial value is:
: 363      1198 1      (XX'FFFFFFFF',XX'7FFFFFFFFF'),    ! -futurest describable date
: 364      1199 1
: 365      1200 1 SINCE_DATE:      ! Selection cutoff date
: 366      1201 1      VECTOR [2, LONG]      ! -allocate one quadword
: 367      1202 1      INITIAL (0,0),      ! -default = oldest describable date
: 368      1203 1
: 369      1204 1 LAST_DATE:      ! Last date encountered
: 370      1205 1      VECTOR [2, LONG]      ! -allocate one quadword
: 371      1206 1      INITIAL (0,0),      ! -default = oldest describable date
: 372      1207 1
: 373      1208 1 FIRST_DATE:      ! Earliest date encountered
: 374      1209 1      BLOCKVECTOR [2, LONG, BYTE]      ! -allocate one quadword
: 375      1210 1      INITIAL      ! -initial value is:
: 376      1211 1      (XX'FFFFFFFF',XX'7FFFFFFFFF'),    ! -futurest describable date
: 377      1212 1
: 378      1213 1
: 379      1214 1
: 380      1215 1 ! POSITION DEPENDENT DATA
: 381      1216 1 ! The next two items, SORT_KEY_COUNT and SORT_TABLE must be
: 382      1217 1 ! in the given order. They are passed as a unit to the sort
: 383      1218 1 !
: 384      1219 1
: 385      1220 1 SORT_KEY_COUNT: WORD,      ! Number of sort keys
: 386      1221 1 SORT_TABLE:      ! Sort key list
: 387      1222 1      BLOCKVECTOR [MAX_SORT+1, QUAD, BYTE],
: 388      1223 1 SORT_INDEX_TABLE:      ! Sort key index table (each byte
: 389      1224 1      VECTOR [MAX_SORT+1, BYTE];      ! contains the index number in the
: 390      1225 1      ! SORT_KEYWORD table for the key).
: 391      1226 1
: 392      1227 1 OWN
: 393      1228 1      SUMMARY_UIC;      ! Flag if /SUMMARY=UIC specified.
: 394      1229 1
: 395      1230 1 $init_state (uic_stb, uic_ktb);      ! UIC parse state table
: 396      1231 1
: 397      P 1232 1 $state (
: 398      P 1233 1      (tpa$_ident,...,CONVERTED_UIC)
: 399      1234 1      );
: 400      P 1235 1 $state (
: 401      P 1236 1      (tpa$_eos, tpa$_exit)
: 402      1237 1      );
```

```

404 1238 1 ROUTINE ACC =
405 1239 1 ! Main routine for ACC
406 1240 1
407 1241 1 ++
408 1242 1 Functional description
409 1243 1 This is the top level routine for the ACC facility.
410 1244 1 It calls the main control loop. Any errors encountered
411 1245 1 will be passed back to this routine.
412 1246 1
413 1247 1 Calling sequence
414 1248 1
415 1249 1 ACC () from the command language interpreter
416 1250 1
417 1251 1 Input parameters
418 1252 1
419 1253 1 AP = Address of argument list passed from CLI
420 1254 1
421 1255 1 Output parameters
422 1256 1
423 1257 1 None
424 1258 1
425 1259 1 Routine value
426 1260 1
427 1261 1 Worst error is returned.
428 1262 1
429 1263 1 ----
430 1264 1
431 1265 2 BEGIN
432 1266 2
433 1267 2 EXTERNAL worst_error,
434 1268 2
435 1269 2 LOCAL channel, status;
436 1270 2
437 1271 2
438 1272 2 SET UP HANDLERS ---
439 1273 2 Declare condition handler to record severest
440 1274 2 error message issued, to be returned on exit of image.
441 1275 2
442 1276 2
443 1277 2 ENABLE handler;
444 1278 2
445 1279 2
446 1280 2
447 1281 2
448 1282 2 CALL MAIN CONTROL ---
449 1283 2 Invoke the main subroutine. If any errors are encountered they
450 1284 2 will be returned immediately, if fatal, or saved in WORST_ERROR
451 1285 2 for exit processing.
452 1286 2
453 1287 2
454 1288 2 Perform (acc_control ());
455 1289 2
456 1290 2
457 1291 2
458 1292 2 RETURN TO USER ---
459 1293 2 Return to the user. Variable WORST_ERROR is maintained by
460 1294 2 the error handler (see routine HANDLER). If no messages have
```

```

: 461      1295 2  !      been signaled then the initial value of WORST_ERROR, SS$_NORMAL,
: 462      1296 2  !      will be returned.
: 463      1297 2  !
: 464      1298 2  !
: 465      1299 2  Return .worst_error;      ! Return contents of WORST_ERROR
: 466      1300 2
: 467      1301 1 END;
```

```

                                .TITLE  ACC
                                .IDENT  \V04-000\
                                .PSECT  _LIB$STATES,NOWRT,  SHR,  PIC,1
                                00000 UIC_STB::
                                      .BLKB  0
                                45EC 00000 :TPASTYPE
                                      U.2: .WORD  17900
                                00000000* 00002 :TPASADDR
                                      U.3: .LONG  <<CONVERTED_UIC-U.3>-4>
                                15F7 00006 :TPASTYPE
                                      U.4: .WORD  5623
                                FFFF 00008 :TPASTARGET
                                      U.5: .WORD  -1
                                .PSECT  _LIB$KEYO$,NOWRT,  SHR,  PIC,1
                                00000 UIC_KTB::
                                      .BLKB  0
                                00000 :TPASKEYO
                                      U.1: .BLKB  0
                                .PSECT  DATA,NOEXE,2
                                00000 KEY_BUF::
                                      .BLKB  1024
                                00400 HOLD_BIOCNT::
                                      .BLKB  4
                                00404 HOLD_DIOCNT::
                                      .BLKB  4
                                00408 HOLD_ELAPSED::
                                      .BLKB  8
                                00410 HOLD_EXECUTION::
                                      .BLKB  4
                                00414 HOLD_PAGEFLTS::
                                      .BLKB  4
                                00418 HOLD_GETCNT::
                                      .BLKB  4
                                0041C HOLD_QIOCNT::
                                      .BLKB  4
                                00420 HOLD_PAGCNT::
                                      .BLKB  4
                                00424 HOLD_PGFLPEAK::
                                      .BLKB  4
                                00428 HOLD_PAGEREADS::
                                      .BLKB  4
                                0042C HOLD_WSPEAK::
```

K 4
15-Sep-1984 23:36:11
14-Sep-1984 11:51:58VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[ACC.SRC]ACC.B32;1Page 12
(5)AC
V0

00430	HOLD_CPU	IM::	.BLKB	4
00438	HOLD_VOLUMES::	.BLKB	8	
0043C	CUSTOMER::	.BLKB	4	
00440	RECORD_LENGTH::	.BLKB	4	
00444	RECORD_SUBTYPE::	.BLKB	4	
00448	RECORD_TYPE::	.BLKB	4	
0044C	VERSION::	.BLKB	4	
00450	GROUP::	.BLKB	4	
00454	MEMBER::	.BLKB	4	
00458	CONVERTED_UIC::	.BLKB	4	
0045C	QUAD_CPU_TIME::	.BLKB	8	
00464	ELAPSED_TIME::	.BLKB	8	
0046C	WAIT_TIME::	.BLKB	8	
00474	ACCOUNT_DESC::	.BLKB	8	
0047C	ADDRESS_DESC::	.BLKB	8	
00484	BUFFERED_IO_DESC::	.BLKB	8	
0048C	CPU_TIME_DESC::	.BLKB	8	
00494	DIRECT_IO_DESC::	.BLKB	8	
0049C	ENTRY_DESC::	.BLKB	8	
004A4	ELAPSED_DESC::	.BLKB	8	
004AC	EXECUTION_DESC::	.BLKB	8	
004B4	FILE_DESC::	.BLKB	8	
004BC	FINISH_TIME_DESC::	.BLKB	8	
004C4	IDENT_DESC::	.BLKB	8	
004CC	IMAGE_FILE_DESC::	.BLKB	8	
004D4	IMAGE_DESC::	.BLKB	8	
004DC	JOB_DESC::	.BLKB	8	
004E4	NODE_DESC::	.BLKB	8	
004EC	OWNER_DESC::	.BLKB	8	

15-Sep-1984 23:36:11
14-Sep-1984 11:51:58

15-Sep-1984 23:36:11

14-Sep-1984 11:51:58

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[ACC.SRC]ACC.B32;1

VAX-11 BLISS-32 V4.0-742

DISK\$VMSMASTER:[ACC.SRC]ACC.B32:1

Page 13
(5)

Page 13

(5)

ACI
VO4

V04

```
004F4 PAGE_FAULTS_DESC:::      .BLKB 8
004FC PAGE_FILE_DESC:::        .BLKB 8
00504 PAGE_READS_DESC:::       .BLKB 8
0050C PRIORITY_DESC:::         .BLKB 8
00514 PRIVILEGE_DESC:::        .BLKB 8
0051C PROCESS_DESC:::          .BLKB 8
00524 QUEUE_DESC:::            .BLKB 8
0052C QUEUE_TIME_DESC:::       .BLKB 8
00534 REMOTE_ID_DESC:::        .BLKB 8
0053C START_TIME_DESC:::       .BLKB 8
00544 STATUS_DESC:::           .BLKB 8
0054C STATUS_TEXT_DESC:::      .BLKB 8
00554 SUBTYPE_DESC:::          .BLKB 8
0055C SYM_GET_DESC:::          .BLKB 8
00564 SYM_QIO_DESC:::          .BLKB 8
0056C SYM_PAGE_DESC:::         .BLKB 8
00574 TERMINAL_DESC:::         .BLKB 8
0057C TYPE_DESC:::             .BLKB 8
00584 TYPE_TEXT_DESC:::        .BLKB 8
0058C UIC_BINARY_DESC:::       .BLKB 8
00594 UIC_GROUP_DESC:::        .BLKB 8
0059C UIC_MEMBER_DESC:::       .BLKB 8
005A4 UIC_DESC:::              .BLKB 8
005AC USER_DATA_DESC:::        .BLKB 8
005B4 USER_DESC:::             .BLKB 8
005BC VOLUMES_DESC:::         .BLKB 8
005C4 WORKING_SET_DESC:::      .BLKB 8
005CC ACCOUNT_SWITCH:::        .BLKB 4
005D0 ADDRESS_SWITCH:::
```

M 4
15-Sep-1984 23:36:11
14-Sep-1984 11:51:58VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[ACC.SRC]ACC.B32;1

	005D4	ENTRY_SWITCH::	.BLKB	4
	005D8	IDENT_SWITCH::	.BLKB	4
	005DC	IMAGE_SWITCH::	.BLKB	4
	005E0	JOB_SWITCH::	.BLKB	4
	005E4	NODE_SWITCH::	.BLKB	4
	005E8	OWNER_SWITCH::	.BLKB	4
	005EC	PRIORITY_SWITCH::	.BLKB	4
	005F0	PROCESS_SWITCH::	.BLKB	4
	005F4	QUEUE_SWITCH::	.BLKB	4
	005F8	REMOTE_ID_SWITCH::	.BLKB	4
	005FC	STATUS_SWITCH::	.BLKB	4
	00600	TERMINAL_SWITCH::	.BLKB	4
	00604	TYPE_SWITCH::	.BLKB	4
	00608	UIC_SWITCH::	.BLKB	4
	0060C	UIC_GROUP_SWITCH::	.BLKB	4
	00610	UIC_MEMBER_SWITCH::	.BLKB	4
	00614	USER_SWITCH::	.BLKB	4
	00618	SELECTED::	.BLKB	4
	0061C	TOTAL_SELECTED::	.BLKB	4
	00620	TOTAL_REJECTED::	.BLKB	4
	00624	SORT_KEY_SIZE::	.BLKB	4
	00628	DATE_HEADER::	.BLKB	4
	0062C	STATUS_TEXT_BUFFER::	.BLKB	80
	0067C	UIC_BUF::	.BLKB	67
	006BF		.BLKB	1
	006C0	A_DATE_BUF::	.BLKB	23
	006D7		.BLKB	1
00000017	006D8	A_DATE_DESC::	.LONG	23
00000000	006DC	ADDRESS A_DATE_BUF		
	006E0	NUMERIC_DATE::		

:


```
20 20 20 20 20 20 20 20 20 20 20 20
006EE .BLKB 14
006F0 S_DATE_BUF: .BLKB 2
006FA .ASCII 1
006FC S_DATE_DESC: .BLKB 2
0000000A 00700 .LONG 10
00000000' 00700 .ADDRESS S_DATE_BUF
00000004 00704 YEAR_DESC: .LONG 4
00000000' 00708 .ADDRESS S_DATE_BUF
00000002 0070C MONTH_DESC: .LONG 2
00000000' 00710 .ADDRESS S_DATE_BUF+5
00000002 00714 DAY_DESC: .LONG 2
00000000' 00718 .ADDRESS S_DATE_BUF+8
0071C WEEKDAY: .BLKB 4
00000001 00720 WEEKDAY_DESC: .LONG 1
00000000' 00724 .ADDRESS WEEKDAY
00000002 00728 HOUR_DESC: .LONG 2
00000000' 0072C .ADDRESS A_DATE_BUF+12
00730 MESSAGE_DESC: .BLKB 8
00# 00738 REPORT_HDR1_FAO: .BYTE 0[3]
02 00738 .BYTE 2
0073C .BLKB 4
00# 00740 REPORT_HDR2_FAO: .BYTE 0[3]
02 00743 .BYTE 2
00744 .BLKB 4
00# 00748 REPORT_HDR3_FAO: .BYTE 0[3]
02 00748 .BYTE 2
0074C .BLKB 4
00# 00750 REPORT_DET_FAO: .BYTE 0[3]
02 00753 .BYTE 2
00754 .BLKB 4
00758 REPORT_VALUE: .BLKB 120
007D0 REPORT_ITEMS: .BLKB 4
007D4 BUCKET_SIZE: .BLKB 4
00# 007D8 SUM_KEY_FAO: .BYTE 0[3]
02 007D8 .BYTE 2
007DC .BLKB 4
007E0 SUM_KEY_VALUE: .BLKB 120
00858 DATE_DESC_ADDR: .BLKB 4
```

```
7FFFFFFF FFFFFFFF 0085C BEFORE_DATE::
                                .LONG -1, 2147483647
00000000 00000000 00864 SINCE_DATE::
                                .LONG 0, 0
00000000 00000000 0086C LAST_DATE::
                                .LONG 0, 0
7FFFFFFF FFFFFFFF 00874 FIRST_DATE::
                                .LONG -1, 2147483647
                                0087C SORT_KEY_COUNT::
                                    .BLKB 2
                                0087E
                                    .BLKB 2
                                00880 SORT_TABLE::
                                    .BLKB 88
                                008D8 SORT_INDEX_TABLE::
                                    .BLKB 11
                                008E3
                                    .BLKB 1
                                008E4 SUMMARY_UIC:
                                    .BLKB 4
```

```
.EXTRN ACC$ INVACC REC, ACC$ TOTAL
.EXTRN ACC$ MERGE, ACC$ INPUT
.EXTRN ACC$ TITLE TRUNC
.EXTRN ADD SYMBOL, ALLOCATE
.EXTRN FIND WATERMARK, HANDLER
.EXTRN LIB$ ADDX, LIB$ CVT DTB
.EXTRN LIB$ CVT HTB, LIB$ CVT TIME
.EXTRN LIB$ DAY, LIB$ FILE SCAN
.EXTRN LIB$ ICHAR, LIB$ LOOKUP KEY
.EXTRN LIB$ SUBX, LIB$ SYS ASC TIM
.EXTRN LIB$ SYS FAO, LIB$ SYS FAOL
.EXTRN LIB$ TPARSE, LOG FILE NAME
.EXTRN LOOKUP SYMBOL, MAP QUALIFIERS
.EXTRN PARSE OUTPUT FILES
.EXTRN RELEASE TO SORT
.EXTRN SCAN SYMBOLS, SHOW RECORD
.EXTRN SOR$ END SORT, SOR$ INIT_SORT
.EXTRN SOR$ RELEASE REC
.EXTRN SOR$ RETURN REC, SOR$ SORT_MERGE
.EXTRN STR$ APPEND, STR$ COMPARE
.EXTRN STR$ DUPL CHAR, STR$ LEFT
.EXTRN STR$ PREFIX, STR$ REPLACE
.EXTRN STR$ RIGHT, STRIP NEGATOR
.EXTRN STRIP TRAIL, SUMMARIZE
.EXTRN SYSS$ NOMTIM, TRANSLATE_STATUS
.EXTRN WRITE BAR GRAPH
.EXTRN WRITE_BINARY, WRITE_SUMMARY
.EXTRN WRITE_TOTALS, INPUT_NAM
.EXTRN INPUT_FAB, INPUT_RAB
.EXTRN OUTPUT_FAB, OUTPUT_RAB
.EXTRN REJECTED_FAB, REJECTED_RAB
.EXTRN WORST_ERROR
```

```
.PSECT CODE, NOWRT, 2
```

```
0000V 6D 0011 0000 00000 ACC:
CF DE 00002
00 FB 00007
```

```
.WORD Save nothing
MOVAL 2$, (FP)
CALLS #0, ACC_CONTROL
```

```
: 1238
: 1265
: 1288
```

ACC
V04-000

C 5
15-Sep-1984 23:36:11 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 11:51:58 DISK\$VMSMASTER:[ACC SRC]ACC.B32,1

Page 17
(5)

07	50	E9	0000C	BLBC	STATUS, 1\$	:	
50	00000000G	00	D0 0000F	MOVL	WORST_ERROR, R0	:	1299
			04 00016	RET		:	1301
			0000 00017	.WORD	Save nothing	:	1265
		7E	D4 00019	CLRL	-(SP)	:	
		5E	DD 0001B	PUSHL	SP	:	
	7E	04	AC 7D 0001D	MOVQ	4(AP), -(SP)	:	
00000000G	00	03	FB 00021	CALLS	#3, HANDLER	:	
		04	00028	RET		:	

; Routine Size: 41 bytes, Routine Base: CODE + 0000

```
469 1302 1 ROUTINE ACC_CONTROL = . Main control routine
470 1303 1
471 1304 1 ++
472 1305 1 Functional description
473 1306 1
474 1307 1 This is the main control routine for the ACC facility.
475 1308 1 It calls the main loop. Any errors encountered
476 1309 1 will be passed back to this routine.
477 1310 1
478 1311 1 Calling sequence
479 1312 1
480 1313 1 ACC_CONTROL ()
481 1314 1
482 1315 1 Input parameters
483 1316 1
484 1317 1 AP = Address of argument list passed from CLI
485 1318 1
486 1319 1 Output parameters
487 1320 1
488 1321 1 None
489 1322 1
490 1323 1 Routine value
491 1324 1
492 1325 1 Worst error is returned.
493 1326 1
494 1327 1 ----
495 1328 1
496 1329 2 BEGIN
497 1330 2
498 1331 2 EXTERNAL
499 1332 2 worst_error; ! Worst error status
500 1333 2
501 1334 2 LOCAL
502 1335 2 temp_file_count, ! Place to save input file count
503 1336 2 desc: vector [2, long], ! Work descriptor
504 1337 2 scan_context: initial( 0 ), ! LIB$FILE_SCAN sticky filespec context
505 1338 2 status; ! Local variable used as a "catch-all"
506 1339 2 ! bucket for R0 values returned by
507 1340 2 ! called procedures
508 1341 2
509 1342 2 OWN
510 1343 2 input_desc: bblock [dsc$k_d_bln] ! Allocate dynamic descriptor
511 1344 2 preset([dsc$b_class] = dsc$k_class_d);
512 1345 2
513 1346 2
514 1347 2
515 1348 2 !
516 1349 2 PROCESS THE COMMAND LINE ---
517 1350 2 Call CLI interface routines to parse the command line. Set up
518 1351 2 internal tables as needed for further processing
519 1352 2
520 1353 2
521 1354 2 Perform (parse_command ()); ! Analyze the command line
522 1355 2
523 1356 2
524 1357 2
525 1358 2
```

```

526      1359 2 | PARSE INPUT FILES --
527      1360 2 |
528      1361 2 |     Get the list of input files from the CLI. Call FILE_SCAN
529      1362 2 |     with each input file name. FILE_SCAN will handle wild-carding
530      1363 2 |     and stickiness and call PROCESS_FILE for each matching file
531      1364 2 |     that is found.
532      1365 2 |
533      1366 2 | While GET VALUE ('INPUT', input_desc) do
534      1367 2 | BEGIN
535      1368 2 |     input_fab [fab$b_fns] = .input_desc [dsc$w_length];
536      1369 2 |     input_fab [fab$l_fna] = .input_desc [dsc$a_pointer];
537      1370 2 |     input_fab [fab$l_ctx] = msg$_searchfail; ! Load the error message
538      1371 2 |     lib$file_scan (                               ! Call file scanner with
539      1372 2 |         input_fab,                                ! -address of FAB
540      1373 2 |         process_file,                             ! -success action routine
541      1374 2 |         log_filename,                             ! -error action routine
542      1375 2 |         scan_context);                            ! -file scan context
543      1376 2 | END;
544      1377 2 |
545      1378 2 |
546      1379 2 | SAVE INPUT FILE COUNT --
547      1380 2 | The input file count is simply the highest value of the file
548      1381 2 | number field that is prepended to each record to be carried
549      1382 2 | through the sort. Save the contents of this field prior to
550      1383 2 | beginning the sort.
551      1384 2 |
552      1385 2 |
553      1386 2 |
554      1387 2 |
555      1388 2 | BEGIN
556      1389 2 | BIND    input_file_count = key_buf + .sort_key_size;
557      1390 2 |          temp_file_count = .input_file_count;           ! Save input file count
558      1391 2 | END;
559      1392 2 |
560      1393 2 | If no files were selected, then simply return.
561      1394 2 |
562      1395 2 | IF .temp_file_count EQLU 0
563      1396 2 | THEN
564      1397 2 | RETURN TRUE;
565      1398 2 |
566      1399 2 | COMPLETE THE SORT --
567      1400 2 | If sort processing is active then trigger the merge phase with
568      1401 2 | a call to SOR$SORT_MERGE. When the merge reaches the final pass,
569      1402 2 | get the sorted records with calls to SOR$RETURN_REC and dispatch
570      1403 2 | them to the output routines.
571      1404 2 |
572      1405 2 |
573      1406 2 |
574      1407 2 |
575      1408 2 |
576      1409 2 | If PRESENT (SORT) then                      ! If sorting is active
577      1410 2 | BEGIN
578      1411 2 | BIND input_buf = key_buf + rec_prefix;       ! Locate record
579      1412 2 | LOCAL temp;                                  ! Sort insists on a bucket
580      1413 2 |
581      1414 2 |
582      1415 2 |

```

```
583 1416 3
584 1417 3 if PRESENT (LOG) then signal (acc$merge, 1, .total_selected);
585 1418 3
586 1419 3 perform (sor$sort_merge ()); ! Do merge phases
587 1420 3
588 1421 3 desc [0] = 512; ! Initialize descriptor
589 1422 3 desc [1] = key_buf,
590 1423 3
591 1424 3 while status = sor$return_rec (desc, temp) ! While input remains
592 1425 3 do BEGIN
593 1426 3 if PRESENT (BINARY) then
594 1427 3 perform (write_binary (input_buf, output_rab))
595 1428 3 ELSE BEGIN
596 1429 3 perform (process_record (input_buf)); ! -handle formats
597 1430 3 perform (show_record (input_buf, output_rab)); ! -call output routines
598 1431 3 END;
599 1432 3 END;
600 1433 3
601 1434 3 if not (.status eql ss$endoffile) then ! If not normal end then
602 1435 3 signal (.status); ! signal the user
603 1436 3
604 1437 3 perform (sor$end_sort ()); ! End the sort
605 1438 3 END;
606 1439 3
607 1440 3
608 1441 3 SET UP REPORT --
609 1442 3 Establish the "from" "to" dates for the report. If explicit
610 1443 3 dates were specified then use them, else allow the earliest
611 1444 3 and latest dates incountered stand.
612 1445 3
613 1446 3
614 1447 3 If PRESENT (SINCE) then MOVE_QUAD (since_date, first_date);
615 1448 3
616 1449 3 If PRESENT (BEFORE) then MOVE_QUAD (before_date, last_date);
617 1450 3
618 1451 3
619 1452 3
620 1453 3
621 1454 3 DO SUMMARY AND TOTALS --
622 1455 3 If /SUMMARY was specified then call the routines that
623 1456 3 output the summary report and a totals page.
624 1457 3
625 1458 3
626 1459 3 If PRESENT (SUMMARY) ! If /SUMMARY
627 1460 3 then if PRESENT (BAR_GRAPH) ! If bar graphs
628 1461 3 then BEGIN
629 1462 3 perform (scan_symbols (
630 1463 3 summation_table, find_watermark)); ! Find all high watermarks
631 1464 3 ! incru i to .report_items do
632 1465 3 perform (scan_symbols ( ! Graph each summed field
633 1466 3 summation_table, write_bar_graph));
634 1467 3 END
635 1468 3 else perform (scan_symbols ! Call write summary for
636 1469 3 summation_table, write_summary); ! each summation record
637 1470 3 Perform (write_totals ()); ! Write a totals page
638 1471 3
639 1472 3
```

```

640 1473 2
641 1474 2
642 1475 2 LOG RESULTS --
643 1476 2     If logging was requested and the saved input file count is
644 1477 2     greater than 1 then display the totals message.
645 1478 2
646 1479 2
647 1480 2
648 1481 2 if PRESENT (LOG) and
649 1482 2     .temp_file_count gtry 1 then
650 1483 2     signal (acc$ total, 3,
651 1484 2     .total_selected,
652 1485 2     .total_rejected,
653 1486 2     .temp_file_count);
654 1487 2
655 1488 2
656 1489 2
657 1490 2
658 1491 2 CLOSE OUTPUT FILES --
659 1492 2     Assign the error messages to be used and close the output
660 1493 2     files. The files are closed only if they have been created.
661 1494 2     Creation is inferred if contents of stream pointers is non-zero.
662 1495 2
663 1496 2
664 1497 2
665 1498 2 Output_fab [fab$l ctx] = msg$ closeout;
666 1499 2 Rejected_fab [fab$l ctx] = msg$ closeout;
667 1500 2
668 1501 2 If PRESENT (OUTPUT) and PRESENT (BINARY)
669 1502 2     then perform ($close (
670 1503 2         fab = output_fab,
671 1504 2         err = log_filename));
672 1505 2
673 1506 2 If PRESENT (REJECTED)
674 1507 2     then perform ($close (
675 1508 2         fab = rejected_fab,
676 1509 2         err = log_filename));
677 1510 2
678 1511 2
679 1512 2
680 1513 2 RETURN TO USER ---
681 1514 2     Return to the user. Variable WORST_ERROR is maintained by
682 1515 2     the error handler (see routine HANDLER). If no messages have
683 1516 2     been signaled then the initial value of WORST_ERROR, $$$_NORMAL,
684 1517 2     will be returned.
685 1518 2
686 1519 2
687 1520 2 Return .worst_error;
688 1521 2
689 1522 1 END;
```

```

00 00 00 54 55 50 4E 49 00029 .BLKB 3
00000005 0002C P.AAB: .ASCII \INPUT\<0><0><0>
00000000 00034 P.AAA: .LONG 5
00000000 00038 .ADDRESS P.AAB
```

```
.PSECT $OWNS,NOEXE,2

00# 00000 INPUT_DESC:
02 00003 .BYTE 0[3]
00004 .BLKB 4

INPUT_BUF= KEY_BUF+8
.EXTRN CLISGET_VALUE, QUALIFIERS
.EXTRN SYSSCLOSE

.PSECT CODE,NOWRT,2

OFFC 00000 ACC_CONTROL:
.SAVE R2,R3,R4,R5,R6,R7,R8,R9,R10,R11
MOVAB SYSSCLOSE, R11
SCAN SYMBOLS, R10
OUTPUT_RAB, R9
LIB$SIGNAL, R8
LOG_FILENAME, R7
INPUT_FAB+52, R6
QUALIFIERS, R5
INPUT_BUF, R4
SUBL2 #12, SP
CLRL SCAN_CONTEXT
CALLS #0, PARSE_COMMAND
BLBC STATUS, 5$
PUSHAB INPUT_DESC
PUSHAB P.AAA
CALLS #2, CLISGET_VALUE
BLBC R0, 2$
MOVAB INPUT_DESC, INPUT_FAB+52
MOVL INPUT_DESC+4, INPUT_FAB+44
MOVL #10424890, INPUT_FAB+24
PUSHR #M<R7,SP>
PUSHAB PROCESS_FILE
PUSHAB INPUT_FAB
CALLS #4, LIB$FILE_SCAN
BRB 1$
MOVAB KEY_BUF, R0
ADDL2 SORT_KEY_SIZE, R0
MOVL (R0), TEMP_FILE_COUNT
BNEQ 3$
MOVL #1, R0
RET
TSTB QUALIFIERS+2
BGEQ 12$
BBC #2, QUALIFIERS+1, 4$
PUSHL TOTAL_SELECTED
PUSHL #1
PUSHL #ACC$ MERGE
CALLS #3, LIB$SIGNAL
CALLS #0, SORT_MERGE
BLBC STATUS, 11$
MOVZWL #512, DESC
MOVAB KEY_BUF, DESC+4
```

5B	00000000G	00	9E	00002	MOVAB	SYSSCLOSE, R11	1382	
5A	00000000G	00	9E	00009	MOVAB	SCAN SYMBOLS, R10		
59	00000000G	00	9E	00010	MOVAB	OUTPUT_RAB, R9		
58	00000000G	00	9E	00017	MOVAB	LIB\$SIGNAL, R8		
57	00000000G	00	9E	0001E	MOVAB	LOG_FILENAME, R7		
56	00000000G	00	9E	00025	MOVAB	INPUT_FAB+52, R6		
55	00000000G	00	9E	0002C	MOVAB	QUALIFIERS, R5		
54	00000000G	EF	9E	00033	MOVAB	INPUT_BUF, R4		
5E		0C	C2	0003A	SUBL2	#12, SP		
		7E	D4	0003D	CLRL	SCAN_CONTEXT	1329	
0000V	CF	00	FB	0003F	CALLS	#0, PARSE_COMMAND	1354	
	6A	50	E9	00044	BLBC	STATUS, 5\$		
		CF	9F	00047	PUSHAB	INPUT_DESC	1367	
		AA	AF	9F	PUSHAB	P.AAA		
00000000G	00	02	FB	0004E	CALLS	#2, CLISGET_VALUE		
	27	50	E9	00055	BLBC	R0, 2\$		
	66	CF	90	00058	MOVAB	INPUT_DESC, INPUT_FAB+52	1369	
F8	A6	CF	D0	0005D	MOVL	INPUT_DESC+4, INPUT_FAB+44	1370	
E4	A6	8F	D0	00063	MOVL	#10424890, INPUT_FAB+24	1371	
		8F	BB	0006B	PUSHR	#M<R7,SP>	1372	
		CF	9F	0006F	PUSHAB	PROCESS_FILE		
		A6	9F	00073	PUSHAB	INPUT_FAB		
00000000G	00	04	FB	00076	CALLS	#4, LIB\$FILE_SCAN		
		C8	11	0007D	BRB	1\$	1367	
	50	A4	9E	0007F	MOVAB	KEY_BUF, R0	1390	
	50	C4	C0	00083	ADDL2	SORT_KEY_SIZE, R0	1391	
	53	60	D0	00088	MOVL	(R0), TEMP_FILE_COUNT	1392	
		04	12	0008B	BNEQ	3\$	1398	
	50	01	D0	0008D	MOVL	#1, R0	1400	
		04	00	00090	RET			
		A5	95	00091	TSTB	QUALIFIERS+2	1412	
		7E	18	00094	BGEQ	12\$		
OF	01	A5	02	E1	00096	BBC	#2, QUALIFIERS+1, 4\$	1417
		C4	DD	0009B	PUSHL	TOTAL_SELECTED		
		01	DD	0009F	PUSHL	#1		
		8F	DD	000A1	PUSHL	#ACC\$ MERGE		
00000000G	68	03	FB	000A7	CALLS	#3, LIB\$SIGNAL		
	00	00	FB	000AA	CALLS	#0, SORT_MERGE	1419	
	5D	50	E9	000B1	BLBC	STATUS, 11\$		
	08	AE	3C	000B4	MOVZWL	#512, DESC	1421	
	0C	AE	F8	9E	000BA	MOVAB	KEY_BUF, DESC+4	1422

		04	AE	9F	000BF	6\$:	PUSHAB	TEMP	1424
		0C	AE	9F	000C2		PUSHAB	DESC	
	00000000G	00	02	FB	000C5		CALLS	#2, SOR\$RETURN_REC	
		52	50	D0	000CC		MOVL	R0, STATUS	
		2A	52	E9	000CF		BLBC	STATUS, 9\$	
0D		65	03	E1	000D2		BBC	#3, QUALIFIERS, 7\$	1426
			8F	BB	000D6		PUSHR	#*M<R4,R9>	1427
	00000000G	00	02	FB	000DA		CALLS	#2, WRITE_BINARY	
			15	11	000E1		BRB	8\$	
			54	DD	000E3	7\$:	PUSHL	R4	1429
	0000V	CF	01	FB	000E5		CALLS	#1, PROCESS_RECORD	
		71	50	E9	000EA		BLBC	STATUS, 18\$	
			8F	BB	000ED		PUSHR	#*M<R4,R9>	1430
	00000000G	00	02	FB	000F1		CALLS	#2, SHOW_RECORD	
		C4	50	E8	000F8	8\$:	BLBS	STATUS, 8\$	
				04	000F9		RET		
	00000870	8F	52	D1	000FC	9\$:	CMPL	STATUS, #2160	1434
			05	13	00103		BEQL	10\$	
			52	DD	00105		PUSHL	STATUS	1435
		68	01	FB	00107		CALLS	#1, LIB\$SIGNAL	
	00000000G	00	00	FB	0010A	10\$:	CALLS	#0, SOR\$END_SORT	1437
		4A	50	E9	00111	11\$:	BLBC	STATUS, 18\$	
07	02	A5	05	E1	00114	12\$:	BBC	#5, QUALIFIERS+2, 13\$	1447
	086C	C4	C4	7D	00119		MOVQ	SINCE DATE, FIRST DATE	
07		65	02	E1	00120	13\$:	BBC	#2, QUALIFIERS, 14\$	1449
	0864	C4	C4	7D	00124		MOVQ	BEFORE DATE, LAST DATE	
		28	A5	E9	0012B	14\$:	BLBC	QUALIFIERS+3, 17\$	1459
16		65	01	E1	0012F		BBC	#1, QUALIFIERS, 15\$	1460
			00	9F	00133		PUSHAB	FIND_WATERMARK	1463
			3F	DD	00139		PUSHL	#63	
		6A	02	FB	0013B		CALLS	#2, SCAN_SYMBOLS	
		79	50	E9	0013E		BLBC	STATUS, 21\$	
			00	9F	00141		PUSHAB	WRITE_BAR_GRAPH	1466
			06	11	00147		BRB	16\$	
			00	9F	00149	15\$:	PUSHAB	WRITE_SUMMARY	1469
			3F	DD	0014F	16\$:	PUSHL	#63	
		6A	02	FB	00151		CALLS	#2, SCAN_SYMBOLS	
		6D	50	E9	00154		BLBC	STATUS, 23\$	
	00000000G	00	00	FB	00157	17\$:	CALLS	#0, WRITE_TOTALS	1470
		63	50	E9	0015E	18\$:	BLBC	STATUS, 23\$	
17	01	A5	02	E1	00161		BBC	#2, QUALIFIERS+1, 19\$	1481
		01	53	D1	00166		CMPL	TEMP_FILE_COUNT, #1	1482
			12	1B	00169		BLEQU	19\$	
			53	DD	0016B		PUSHL	TEMP_FILE_COUNT	1486
		7E	C4	7D	0016D		MOVQ	TOTAL_SELECTED, -(SP)	1484
			03	DD	00172		PUSHL	#3	1483
			8F	DD	00174		PUSHL	#ACCS TOTAL	
		68	05	FB	0017A		CALLS	#5, LIB\$SIGNAL	
	00000000G	00	8F	D0	0017D	19\$:	MOVL	#10424410, OUTPUT FAB+24	1498
	00000000G	00	8F	D0	00188		MOVL	#10424410, REJECTED FAB+24	1499
12		A5	06	E1	00193		BBC	#6, QUALIFIERS+1, 20\$	1501
0E		65	03	E1	00198		BBC	#3, QUALIFIERS, 20\$	
			57	DD	0019C		PUSHL	R7	1504
			00	9F	0019E		PUSHAB	OUTPUT FAB	
		6B	02	FB	001A4		CALLS	#2, SYS\$CLOSE	
		1A	50	E9	001A7		BLBC	STATUS, 23\$	
0E	02	A5	02	E1	001AA	20\$:	BBC	#2, QUALIFIERS+2, 22\$	1506

ACC
V04-000

J 5
15-Sep-1984 23:36:11 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 11:51:58 DISK\$VMSMASTER:[ACC.SRC]ACC.B32;1

Page 24
(6)

A
V

	00000000G	57	DD	001AF	PUSHL	R7	:	1509
		00	9F	001B1	PUSHAB	REJECTED_FAB	:	
6B		02	FB	001B7	CALLS	#2, SYSS\$CLOSE	:	
07		50	E9	001BA	BLBC	STATUS, 23\$	:	
50	00000000G	00	D0	001BD	MOVL	WORST_ERROR, R0	:	1520
		04	001C4	23\$:	RET		:	1522

; Routine Size: 453 bytes, Routine Base: CODE + 003C

```
691 1523 1 ROUTINE PROCESS_FILE (FAB) =
692 1524 1
693 1525 1 ----
694 1526 1
695 1527 1 Functional description
696 1528 1
697 1529 1 This routine processes one input file. It is called as an action
698 1530 1 routine from FILESCAN.
699 1531 1
700 1532 1 Input parameters
701 1533 1
702 1534 1
703 1535 1 Output parameters
704 1536 1
705 1537 1 Record processing routines are called as necessary.
706 1538 1 Any errors encountered are RETURNed immediately.
707 1539 1 TRUE is returned on a normal exit.
708 1540 1
709 1541 1 ----
710 1542 1
711 1543 2 BEGIN
712 1544 2
713 1545 2 MAP
714 1546 2 fab: ref bblock; ! Define as a block structure
715 1547 2
716 1548 2 EXTERNAL
717 1549 2 related_nam: BBLOCK [], ! Related NAM block for input parse
718 1550 2 input_xabfhc: BBLOCK [], ! Input file FHC XAB block
719 1551 2 key_table;
720 1552 2
721 1553 2 BIND
722 1554 2 input_file_count = key_buf + .sort_key_size,
723 1555 2 record_number = key_buf + .sort_key_size + 4,
724 1556 2 input_buf = key_buf + .sort_key_size + rec_prefix;
725 1557 2
726 1558 2 LOCAL
727 1559 2 status: bblock [long], ! Local "catch all" status return
728 1560 2 desc: vector [2, long]; ! Temporary string descriptor
729 1561 2
730 1562 2 OWN
731 1563 2 sort_init: initial (false); ! Sort initialization flag
732 1564 2
733 1565 2
734 1566 2 ESTABLISH THE BUFFER ADDRESS --
735 1567 2
736 1568 2
737 1569 2 input_rab [rab$l_ubf] = input_buf; ! Set input buffer address
738 1570 2
739 1571 2
740 1572 2
741 1573 2 OPEN AND CONNECT ---
742 1574 2 To the input accounting file. Exit immediately if any errors
743 1575 2 are detected. The error handlers will have been invoked if an
744 1576 2 error occurred, and the user will have been notified. The error
745 1577 2 message used by the LOG_FILENAME routine is drawn from the
746 1578 2 CIX field of the FAB or INPUT_RAB as needed.
747 1579 2
```

```

748 1580 2
749 1581 2 fab [fab$l_ctx] = msg$ openin;          ! Load the error mess.
750 1582 2 perform ($open (fab=.fab, err=log_filename)); ! OPEN the input file
751 1583 2 perform ($connect (rab=input_rab, err=log_filename)); ! CONNECT to input file
752 1584 2
753 1585 2
754 1586 2 INITIALIZE THE SORT --
755 1587 2     If sorting was requested then initialize the sort.
756 1588 2
757 1589 2
758 1590 2 (sort_key_count+2)<0,16> = .sort_key_count;      ! *** HACK
759 1591 2
760 1592 2 If PRESENT (SORT) AND (.sort_init eql false) then
761 1593 2 BEGIN
762 1594 2     LOCAL
763 1595 2         lrl, ebk;
764 1596 2         lrl = .input_xabfhc [xab$w_lrl] + .sort_key_size + rec_prefix;
765 1597 2         ebk = .input_xabfhc [xab$l_ebk];
766 1598 2         signal_if_error (sor$init_sort (sort_key_count+2, lrl, ebk));
767 1599 2         sort_init = true;
768 1600 2     END;
769 1601 2
770 1602 2
771 1603 2
772 1604 2 INITIALIZE OUTPUT FILES --
773 1605 2     The processing of the output files has been deferred until now
774 1606 2     so that a fully parsed input file name would be available (as
775 1607 2     a related file name) for default file name components
776 1608 2
777 1609 2
778 1610 2
779 1611 2 If .input_file_count eql 0 then          ! If this is the first pass
780 1612 2     signal_if_error (parse_output_files ()); ! then open the output files
781 1613 2
782 1614 2 input_file_count = .input_file_count + 1; ! Bump the file count.
783 1615 2
784 1616 2
785 1617 2
786 1618 2
787 1619 2
788 1620 2 RESET THE RECORD COUNTERS--
789 1621 2     Reset the file-relative record number such that records in each
790 1622 2     file will be numbered in ascending order beginning with one.
791 1623 2     Reset the selected count so it is also on a per-file basis.
792 1624 2
793 1625 2
794 1626 2 record_number = 0;          ! Reset the record number
795 1627 2 selected = 0;              ! Reset file select count
796 1628 2
797 1629 2
798 1630 2
799 1631 2
800 1632 2
801 1633 2 READ AND LOOP UNTIL EOF ---
802 1634 2     Now read the file until EOF or an error is detected. Process
803 1635 2     each record as specified in the command line
804 1636 2
```

```
1637 2
1638 2 While status = $get (rab=input_rab, err=log_filename) do ! Read a record
1639 3 BEGIN
1640 3 record_number = .record_number + 1; ! Update the record number
1641 3 if not process_record (input_buf) ! Analyze the record
1642 3 then
1643 4 begin
1644 4 desc [0] = .input_nam [nam$b_rsl]; ! then notify the user
1645 4 desc [1] = .input_nam [nam$l_rsa]; ! with file and counts
1646 4 signal( msg$_readerr, 1, desc, acc$_invaccrrec, 1, .record_number );
1647 4 end
1648 3 else
1649 3 if record_selected (input_buf)
1650 3 then
1651 4 if PRESENT (SORT)
1652 4 then
1653 3 signal_if_error (release_to_sort (input_buf, sort_table, key_table,
1654 4 key_buf));
1655 3 else
1656 4 if PRESENT (BINARY)
1657 4 then
1658 4 signal_if_error (write_binary (input_buf, output_rab))
1659 4 else
1660 4 signal_if_error (show_record (input_buf, output_rab))
1661 3 else
1662 4 if PRESENT (REJECTED) ! If secondary stream open
1663 4 then
1664 4 signal_if_error (write_binary (input_buf, rejected_rab));
1665 2 END;
1666 2
1667 2 Fab [fab$l_ctx] = msg$_closein; ! Assign error message
1668 2 Perform ($close (fab=.fab, err=log_filename)); ! Close the input file
1669 2
1670 2 Total_selected = .total_selected + .selected; ! Accumulate totals
1671 2 Total_rejected = .total_rejected + (.record_number - .selected);
1672 2
1673 2 !
1674 2 ! CHECK STATUS AT END OF LOOP ---
1675 2 ! Now check the return status to make sure it was a normal EOF. If not,
1676 2 ! notify the user.
1677 2 !
1678 2 !
1679 3 If not (.status eql rms$_eof) ! If any status other than
1680 3 then return .status; ! expected eof, return it
1681 2 !
1682 2 !
1683 2 CLOSE ---
1684 2 Input file processing is now complete. Revise the stored error
1685 2 message (which is passed to the error routine via the 'user context'
1686 2 field [CTX] of the FAB) and $CLOSE the input file.
1687 2 !
1688 2 !
1689 2 !
1690 2 If PRESENT (LOG) then ! If logging requested
1691 3 BEGIN
1692 3 desc [0] = .input_nam [nam$b_rsl]; ! then notify the user
1693 3 desc [1] = .input_nam [nam$l_rsa]; ! with file and counts
```

```
: 862      1694 3      signal (acc$_input, 3, desc, .selected, .record_number - .selected);  
: 863      1695 2      END;  
: 864      1696 2  
: 865      1697 2      Return true;  
: 866      1698 2  
: 867      1699 1      END;
```

.PSECT \$OWNS,NOEXE,2

00000000 00008 SORT_INIT:

.LONG 0

.EXTRN RELATED_NAM, INPUT_XABFHC

.EXTRN KEY_TABLE, SYSS\$OPEN

.EXTRN SYSS\$CONNECT, SYSS\$GET

.PSECT CODE,NOWRT,2

OFFC 00000 PROCESS_FILE:

	5B	00000000G	00	9E	00002	.JORD	Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11	1523	
	5A	00000000G	00	9E	00009	MOVAB	INPUT_RAB, R11		
	59	00000000G	00	9E	00010	MOVAB	INPUT_NAM+3, R10		
	58	00000000G	00	9E	00017	MOVAB	LOG_FILENAME, R9		
	57	00000000'	EF	9E	0001E	MOVAB	QUALIFIERS, R8		
	5E		10	C2	00025	MOVAB	SORT_KEY_SIZE, R7		
	50		67	D0	00026	SUBL2	#16, SP		
	55	F9DC	C740	9E	0002B	MOVL	SORT_KEY_SIZE, R0	1554	
	56	F9E0	C740	9E	00031	MOVAB	KEY_BUF[R0], R5		
	54	F9E4	C740	9E	00037	MOVAB	KEY_BUF+4[R0], R6	1555	
	24	AB	54	D0	0003D	MOVAB	KEY_BUF+8[R0], R4	1556	
	52	04	AC	D0	00041	MOVL	R4, INPUT_RAB+36	1569	
	18	A2	009F109A	8F	D0	00045	MOVL	FAB, R2	1581
		0204	8F	BB	0004D	MOVL	#10424474, 24(R2)		
00000000G	00		02	FB	00051	PUSHR	#^M<R2,R9>	1582	
	0B		50	E9	00058	CALLS	#2, SYSS\$OPEN		
			59	DD	0005B	BLBC	STATUS, 1\$		
			5B	DD	0005D	PUSHL	R9	1583	
00000000G	00		02	FA	0005F	PUSHL	R11		
	01		50	E4	00066	CALLS	#2, SYSS\$CONNECT		
			04	00069	BLBS	STATUS, 2\$			
025A	C7	0258	C7	B0	0006A	RET			
		02	AB	95	00071	MOVW	SORT_KEY_COUNT, SORT_KEY_COUNT+2	1590	
			37	18	00074	TSTB	QUALIFIERS+2	1592	
		0000'	CF	D5	00076	BGEQ	3\$		
			31	12	0007A	TSTL	SORT_INIT		
	50	00000000G	00	3C	0007C	BNEQ	3\$		
	50		67	C0	00083	MOVZWL	INPUT_XABFHC+10, R0	1596	
04	AE	08	A0	9E	00086	ADDL2	SORT_KEY_SIZE, R0		
	6E	00000000G	00	D0	0008B	MOVAB	8(R0), LRL		
			5E	DD	00092	MOVL	INPUT_XABFHC+16, EBK	1597	
		08	AE	9F	00094	PUSHL	SP	1598	
		025A	C7	9F	00097	PUSHAB	LRL		
00000000G	00		03	FB	0009B	PUSHAB	SORT_KEY_COUNT+2		
	53		50	D0	000A2	CALLS	#3, SORT\$INIT_SORT		
						MOVL	R0, STATUS		

0000'	13	53	E9	000A5	BLBC	STATUS, 4\$	
	CF	01	D0	000A8	MOVL	#1, SORT_INIT	1599
		65	D5	000AD	TSTL	(R5)	1611
00000000G	00	10	12	000AF	BNEQ	5\$	
	53	00	FB	000B1	CALLS	#0, PARSE_OUTPUT_FILES	1612
	03	50	D0	000B8	MOVL	R0, STATUS	
		53	E8	000BB	4\$: BLBS	STATUS, 5\$	
		00AD	31	000BE	BRW	15\$	
		65	D6	000C1	5\$: INCL	(R5)	1614
		66	D4	000C3	CLRL	(R6)	1626
		F4	A7	000C5	CLRL	SELECTED	1627
		59	DD	000C8	6\$: PUSHL	R9	1638
		5B	DD	000CA	PUSHL	R11	
00000000G	00	02	FB	000CC	CALLS	#2, SYS\$GET	
	55	50	D0	000D3	MOVL	R0, STATUS	
	03	55	E8	000D6	BLBS	STATUS, 7\$	
		009F	31	000D9	BRW	16\$	
		66	D6	000DC	7\$: INCL	(R6)	1640
		54	DD	000DE	PUSHL	R4	1641
0000V	CF	01	FB	000E0	CALLS	#1, PROCESS_RECORD	
	27	50	E8	000E5	BLBS	R0, 9\$	
08	AE	6A	9A	000E8	MOVZBL	INPUT_NAM+3, DESC	1644
0C	AE	01	AA	000EC	MOVL	INPUT_NAM+4, DESC+4	1645
		66	DD	000F1	PUSHL	(R6)	1646
		01	DD	000F3	PUSHL	#1	
		00000000G	8F	DD	000F5	PUSHL	#ACCS_INVACCREC
		14	AE	9F	000FB	PUSHAB	DESC
		01	DD	000FE	PUSHL	#1	
00000000G	00	009F10B2	8F	DD	00100	PUSHL	#10424498
		06	FB	00106	CALLS	#6, LIB\$SIGNAL	
		B9	11	0010D	8\$: BRB	6\$	1641
		54	DD	0010F	9\$: PUSHL	R4	1649
0000V	CF	01	FB	00111	CALLS	#1, RECORD_SELECTED	
	3B	50	E9	00116	BLBC	R0, 12\$	
		02	A8	95	00119	TSTB	QUALIFIERS+2
		19	18	0011C	BGEQ	10\$	1651
		F9DC	C7	9F	0011E	PUSHAB	KEY_BUF
		00000000G	00	9F	00122	PUSHAB	KEY_TABLE
		025C	C7	9F	00128	PUSHAB	SORT_TABLE
		54	DD	0012C	PUSHL	R4	
00000000G	00	04	FB	0012E	CALLS	#4, RELEASE_TO_SORT	
		31	11	00135	BRB	14\$	
08	68	03	E1	00137	10\$: BBC	#3, QUALIFIERS, 11\$	1656
		00000000G	00	9F	0013B	PUSHAB	OUTPUT_RAB
		1C	11	00141	BRB	13\$	1658
		00000000G	00	9F	00143	11\$: PUSHAB	OUTPUT_RAB
		54	DD	00149	PUSHL	R4	1660
00000000G	00	02	FB	0014B	CALLS	#2, SHOW_RECORD	
		14	11	00152	BRB	14\$	
B4	02	02	E1	00154	12\$: BBC	#2, QUALIFIERS+2, 8\$	1662
		00000000G	00	9F	00159	PUSHAB	REJECTED_RAB
		54	DD	0015F	13\$: PUSHL	R4	1664
00000000G	00	02	FB	00161	CALLS	#2, WRITE_BINARY	
	53	50	D0	00168	14\$: MOVL	R0, STATUS	
	9F	53	E8	0016B	BLBS	STATUS, 8\$	
		53	DD	0016E	15\$: PUSHL	STATUS	
00000000G	00	01	FB	00170	CALLS	#1, LIB\$SIGNAL	

	50		53	D0 00177	MOVL	STATUS, R0	:	
				04 0017A	RET		:	
18	A2	009F1052	8F	D0 0017B	16\$: MOVL	#10424402, 24(R2)	:	1667
		0204	8F	BB 00183	PUSHR	#^M<R2,R9>	:	1668
00000000G	00		02	F8 00187	CALLS	#2, SYS\$CLOSE	:	
	42		50	E9 0018E	BLBC	STATUS, 19\$	:	
	50	F4	A7	D0 00191	MOVL	SELECTED, R0	:	1670
51	F8		50	C0 00195	ADDL2	R0, TOTAL_SELECTED	:	
	66		50	C3 00199	SUBL3	R0, (R6), R1	:	1671
	FC		51	C0 0019D	ADDL2	R1, TOTAL_REJECTED	:	
0001827A	8F		55	D1 001A1	CMPL	STATUS, #98938	:	1679
			04	13 001A8	BEQL	17\$	:	
	50		55	D0 001AA	MOVL	STATUS, R0	:	1680
				04 001AD	RET		:	
1D	01	A8	02	E1 001AE	17\$: BBC	#2, QUALIFIERS+1, 18\$	:	1690
	08	AE	6A	9A 001B3	MOVZBL	INPUT_NAM+3, DESC	:	1692
	0C	AE	AA	D0 001B7	MOVL	INPUT_NAM+4, DESC+4	:	1693
			03	BB 001BC	PUSHR	#^M<R0,R1>	:	1694
			AE	9F 001BE	PUSHAB	DESC	:	
			03	DD 001C1	PUSHL	#3	:	
		00000000G	8F	DD 001C3	PUSHL	#ACCS INPUT	:	
00000000G	00		05	FB 001C9	CALLS	#5, LIB\$SIGNAL	:	
	50		01	D0 001D0	18\$: MOVL	#1, R0	:	1697
			04	001D3	19\$: RET		:	1699

; Routine Size: 468 bytes. Routine Base: CODE + 0201


```

869 1700 1 ROUTINE PROCESS_RECORD (BUFFER) =
870 1701 1
871 1702 1 ----
872 1703 1
873 1704 1 Functional description
874 1705 1
875 1706 1 This routine is called to analyze a record from the system
876 1707 1 accounting file.
877 1708 1
878 1709 1 Input parameters
879 1710 1
880 1711 1 BUFFER = Address of a buffer containing the record to be detailed.
881 1712 1 The record can be of any type from the V2 and V3 accounting
882 1713 1 formats
883 1714 1
884 1715 1 Output parameters
885 1716 1
886 1717 1 Based on the contents of the record, values are established.
887 1718 1 These values are used in later processing.
888 1719 1 Any errors encountered are RETURNed immediately.
889 1720 1 TRUE is returned on a normal exit.
890 1721 1
891 1722 1 ----
892 1723 1
893 1724 2 BEGIN
894 1725 2
895 1726 2 MAP
896 1727 2 buffer: ref bblock; ! Define buffer format
897 1728 2
898 1729 2 LOCAL
899 1730 2 day_number, ! Temporary output of LIB$DAY
900 1731 2 day_time; ! Temporary output of LIB$DAY
901 1732 2
902 1733 2
903 1734 2
904 1735 2 Initialize descriptors
905 1736 2
906 1737 2 customer = 0;
907 1738 2 record_length = 0;
908 1739 2 record_type = 0;
909 1740 2 record_subtype = 0;
910 1741 2 version = 0;
911 1742 2
912 1743 2 account_desc [0] = 0;
913 1744 2 account_desc [1] = 0;
914 1745 2 address_desc [0] = 0;
915 1746 2 address_desc [1] = 0;
916 1747 2 buffered_io_desc [0] = 0;
917 1748 2 buffered_io_desc [1] = 0;
918 1749 2 cpu_time_desc [0] = 0;
919 1750 2 cpu_time_desc [1] = 0;
920 1751 2 direct_io_desc [0] = 0;
921 1752 2 direct_io_desc [1] = 0;
922 1753 2 elapsed_desc [0] = 0;
923 1754 2 elapsed_desc [1] = elapsed_time;
924 1755 2 elapsed_time [0] = 0;
925 1756 2 elapsed_time [1] = 0;
```

926	1757	2	execution_desc [0]	=	0:
927	1758	2	execution_desc [1]	=	0:
928	1759	2	file_desc [0]	=	0:
929	1760	2	file_desc [1]	=	0:
930	1761	2	finish_time_desc [0]	=	0:
931	1762	2	finish_time_desc [1]	=	0:
932	1763	2	image_file_desc [0]	=	0:
933	1764	2	image_file_desc [1]	=	0:
934	1765	2	image_desc [0]	=	0:
935	1766	2	image_desc [1]	=	0:
936	1767	2	entry_desc [0]	=	0:
937	1768	2	entry_desc [1]	=	0:
938	1769	2	job_desc [0]	=	0:
939	1770	2	job_desc [1]	=	0:
940	1771	2	node_desc [0]	=	0:
941	1772	2	node_desc [1]	=	0:
942	1773	2	owner_desc [0]	=	0:
943	1774	2	owner_desc [1]	=	0:
944	1775	2	page_faults_desc [0]	=	0:
945	1776	2	page_faults_desc [1]	=	0:
946	1777	2	page_file_desc [0]	=	0:
947	1778	2	page_file_desc [1]	=	0:
948	1779	2	page_reads_desc [0]	=	0:
949	1780	2	page_reads_desc [1]	=	0:
950	1781	2	ident_desc [0]	=	0:
951	1782	2	ident_desc [1]	=	0:
952	1783	2	priority_desc [0]	=	0:
953	1784	2	priority_desc [1]	=	0:
954	1785	2	privilege_desc [0]	=	0:
955	1786	2	privilege_desc [1]	=	0:
956	1787	2	process_desc [0]	=	0:
957	1788	2	process_desc [1]	=	0:
958	1789	2	quad_cpu_time [0]	=	0:
959	1790	2	quad_cpu_time [1]	=	0:
960	1791	2	queue_desc [0]	=	0:
961	1792	2	queue_desc [1]	=	0:
962	1793	2	queue_time_desc [0]	=	0:
963	1794	2	queue_time_desc [1]	=	0:
964	1795	2	remote_id_desc [0]	=	0:
965	1796	2	remote_id_desc [1]	=	0:
966	1797	2	start_time_desc [0]	=	0:
967	1798	2	start_time_desc [1]	=	0:
968	1799	2	status_desc [0]	=	0:
969	1800	2	status_desc [1]	=	0:
970	1801	2	status_text_desc [0]	=	0:
971	1802	2	status_text_desc [1]	=	0:
972	1803	2	subtype_desc [0]	=	0:
973	1804	2	subtype_desc [1]	=	0:
974	1805	2	sym_get_desc [0]	=	0:
975	1806	2	sym_get_desc [1]	=	0:
976	1807	2	sym_page_desc [0]	=	0:
977	1808	2	sym_page_desc [1]	=	0:
978	1809	2	sym_qio_desc [0]	=	0:
979	1810	2	sym_qio_desc [1]	=	0:
980	1811	2	terminal_desc [0]	=	0:
981	1812	2	terminal_desc [1]	=	0:
982	1813	2	type_desc [0]	=	0:

```

: 983      1814 2 type_desc [1]          = 0:
: 984      1815 2 type_text_desc [0]     = 0:
: 985      1816 2 type_text_desc [1]     = 0:
: 986      1817 2 uic_desc [0]           = 0:
: 987      1818 2 uic_desc [1]           = 0:
: 988      1819 2 uic_binary_desc [0]     = 0:
: 989      1820 2 uic_binary_desc [1]     = 0:
: 990      1821 2 uic_group_desc [0]      = 0:
: 991      1822 2 uic_group_desc [1]      = 0:
: 992      1823 2 uic_member_desc [0]     = 0:
: 993      1824 2 uic_member_desc [1]     = 0:
: 994      1825 2 user_data_desc [0]      = 0:
: 995      1826 2 user_data_desc [1]      = 0:
: 996      1827 2 user_desc [0]          = 0:
: 997      1828 2 user_desc [1]          = 0:
: 998      1829 2 volumes_desc [0]       = 0:
: 999      1830 2 volumes_desc [1]       = 0:
: 1000     1831 2 wait_time [0]           = 0:
: 1001     1832 2 wait_time [1]           = 0:
: 1002     1833 2 working_set_desc [0]    = 0:
: 1003     1834 2 working_set_desc [1]    = 0:
: 1004     1835 2
: 1005     1836 2 hold_biocnt             = 0:
: 1006     1837 2 hold_diocnt             = 0:
: 1007     1838 2 hold_elapsed [0]        = 0:
: 1008     1839 2 hold_elapsed [1]        = 0:
: 1009     1840 2 hold_execution          = 0:
: 1010     1841 2 hold_pageflts          = 0:
: 1011     1842 2 hold_getcnt            = 0:
: 1012     1843 2 hold_qiocnt            = 0:
: 1013     1844 2 hold_pagcnt            = 0:
: 1014     1845 2 hold_pgflpeak           = 0:
: 1015     1846 2 hold_pagereads          = 0:
: 1016     1847 2 hold_wspeak            = 0:
: 1017     1848 2 hold_cputim [0]         = 0:
: 1018     1849 2 hold_cputim [1]         = 0:
: 1019     1850 2 hold_volumes            = 0:
: 1020     1851 2
: 1021     1852 2 version = .buffer [acr$v_version];
: 1022     1853 2
: 1023     1854 2 SELECTONEU .version of
: 1024     1855 2     SET
: 1025     1856 2     [acr$k_version2]:
: 1026     1857 2         perform (process_v2_record(.buffer));
: 1027     1858 2     [acr$k_version3t, acr$k_version3]:
: 1028     1859 2         perform (process_v3_record(.buffer));
: 1029     1860 2     [OTHERWISE]:
: 1030     1861 2         return false;
: 1031     1862 2     TES;
: 1032     1863 2
: 1033     1864 3 If PRESENT (REPORT)
: 1034     1865 3     THEN BEGIN
: 1035     1866 3         IF .buffered_io_desc [0] neq 0
: 1036     1867 4             THEN BEGIN
: 1037     1868 4                 hold_biocnt = ..buffered_io_desc [1];
: 1038     1869 3             END;
: 1039     1870 3         IF .cpu_time_desc [0] neq 0
```

```
1040 1871 4 THEN BEGIN
1041 1872 4 hold_cpu_time [0] = .quad_cpu_time [0];
1042 1873 4 hold_cpu_time [1] = .quad_cpu_time [1];
1043 1874 3 END;
1044 1875 3 IF .direct_io_desc [0] neq 0
1045 1876 4 THEN BEGIN
1046 1877 4 hold_diocnt = ..direct_io_desc [1];
1047 1878 3 END;
1048 1879 3 hold_elapsed [0] = .elapsed_time [0];
1049 1880 3 hold_elapsed [1] = .elapsed_time [1];
1050 1881 3 perform (lib$day (day_number, hold_elapsed, day_time));
1051 1882 3 hold_elapsed [0] = (.day_time/1000) + (.day_number * 24 * 60 * 60);
1052 1883 3 hold_elapsed [1] = 0;
1053 1884 3 IF .execution_desc [0] neq 0
1054 1885 4 THEN BEGIN
1055 1886 4 hold_execution = ..execution_desc [1];
1056 1887 3 END;
1057 1888 3 IF .page_faults_desc [0] neq 0
1058 1889 4 THEN BEGIN
1059 1890 4 hold_pageflts = ..page_faults_desc [1];
1060 1891 3 END;
1061 1892 3 IF .page_file_desc [0] neq 0
1062 1893 4 THEN BEGIN
1063 1894 4 hold_pgflpeak = ..page_file_desc [1];
1064 1895 3 END;
1065 1896 3 IF .page_reads_desc [0] neq 0
1066 1897 4 THEN BEGIN
1067 1898 4 hold_pagereads = ..page_reads_desc [1];
1068 1899 3 END;
1069 1900 3 IF .sym_get_desc [0] neq 0
1070 1901 4 THEN BEGIN
1071 1902 4 hold_getcnt = ..sym_get_desc [1];
1072 1903 3 END;
1073 1904 3 IF .sym_qio_desc [0] neq 0
1074 1905 4 THEN BEGIN
1075 1906 4 hold_qiocnt = ..sym_qio_desc [1];
1076 1907 3 END;
1077 1908 3 IF .sym_page_desc [0] neq 0
1078 1909 4 THEN BEGIN
1079 1910 4 hold_pagcnt = ..sym_page_desc [1];
1080 1911 3 END;
1081 1912 3 IF .volumes_desc [0] neq 0
1082 1913 4 THEN BEGIN
1083 1914 4 hold_volumes = ..volumes_desc [1];
1084 1915 3 END;
1085 1916 3 IF .working_set_desc [0] neq 0
1086 1917 4 THEN BEGIN
1087 1918 4 hold_wspeak = ..working_set_desc [1];
1088 1919 3 END;
1089 1920 3 END;
1090 1921 3
1091 1922 3 IF PRESENT (SORT) OR PRESENT (EXPAND_DATE)
1092 1923 4 THEN BEGIN
1093 1924 4
1094 1925 4 form an ascii date string in the form:
1095 1926 4 DD-MMM-YYYY HH:MM:SS.CC
1096 1927 3
```

```
1097 1928 3 !
1098 1929 3
1099 1930 3 Perform (lib$sys_asctim (0, a_date_desc, ..date_desc_addr, 0));
1100 1931 3 END;
1101 1932 3
1102 1933 3 If PRESENT (EXPAND_DATE)
1103 1934 3 THEN BEGIN
1104 1935 3
1105 1936 3
1106 1937 3 Calculate day-of-week in binary byte range of 0-6
1107 1938 3 First get total days since epoch date
1108 1939 3 Shift it 3 days to make 0 = Sunday
1109 1940 3 Convert to modulo 7
1110 1941 3 Convert to ascii byte as a temporary hack
1111 1942 3
1112 1943 3
1113 1944 3 perform (lib$day (weekday, ..date_desc_addr));
1114 1945 3
1115 1946 3 weekday = ((.weekday + 3) mod 7) + '0';
1116 1947 3
1117 1948 3
1118 1949 3
1119 1950 3 Form an ascii date string suitable for sorting in the form:
1120 1951 3
1121 1952 3 YYY MM DD
1122 1953 3
1123 1954 3
1124 1955 3 perform (sys$numtim (numeric_date, ..date_desc_addr));
1125 1956 3
1126 1957 3 perform (lib$sys_fao (ad ('!4ZW !2ZW !2ZW'),
1127 P 1958 3 0, ! No output length
1128 P 1959 3 s_date_desc, ! Output buf desc
1129 P 1960 3 .numeric_date [0], ! Year
1130 P 1961 3 .numeric_date [1], ! Month
1131 P 1962 3 .numeric_date [2] ! Day of month
1132 1963 3 ));
1133 1964 2 END;
1134 1965 2
1135 1966 2 return true;
1136 1967 2
1137 1968 1 END;
```

```
00 57 5A 32 21 20 57 5A 32 21 20 57 5A 34 21 003D5 .BLKB 3
00 003D8 P.AAD: .ASCII \!4ZW !2ZW !2ZW\<0><0>
00 003E7
0000000E 003E8 P.AAC: .LONG 14
00000000 003EC .ADDRESS P.AAD
```

```
000C 0000 PROCESS_RECORD:
53 00000000G 00 9E 00002 .WORD Save R2,R3
52 00000000' EF 9E 00009 MOVAB QUALIFIERS, R3
FO A2 7C 00010 MOVAB VERSION, R2
CLRQ CUSTOMER
```

```
: 1700
:
: 1737
```

SC A2

F8	A2	7C	00013	CLRQ	RECORD SUBTYPE	1740
62	D4	00016	CLRL	VERSION	1741	
28	A2	7C	00018	CLRQ	ACCOUNT_DESC	1743
30	A2	7C	0001B	CLRQ	ADDRESS_DESC	1745
38	A2	7C	0001E	CLRQ	BUFFERED_IO_DESC	1747
40	A2	7C	00021	CLRQ	CPU_TIME_DESC	1749
48	A2	7C	00024	CLRQ	DIRECT_IO_DESC	1751
18	A2	9E	00027	MOVAB	ELAPSED_TIME, ELAPSED_DESC+4	1754
18	A2	7C	0002C	CLRQ	ELAPSED_TIME	1755
60	A2	7C	0002F	CLRQ	EXECUTION_DESC	1757
68	A2	7C	00032	CLRQ	FILE_DESC	1759
70	A2	7C	00035	CLRQ	FINISH_TIME_DESC	1761
0080	C2	7C	00038	CLRQ	IMAGE_FILE_DESC	1763
0088	C2	7C	0003C	CLRQ	IMAGE_DESC	1765
50	A2	D4	00040	CLRL	ENTRY_DESC	1767
54	A2	7C	00043	CLRQ	ENTRY_DESC+4	1768
0090	C2	7C	00046	CLRQ	JOB_DESC	1769
0098	C2	7C	0004A	CLRQ	NODE_DESC	1771
00A0	C2	7C	0004E	CLRQ	OWNER_DESC	1773
00A8	C2	7C	00052	CLRQ	PAGE_FAULTS_DESC	1775
00B0	C2	7C	00056	CLRQ	PAGE_FILE_DESC	1777
00B8	C2	7C	0005A	CLRQ	PAGE_READS_DESC	1779
78	A2	7C	0005E	CLRQ	IDENT_DESC	1781
00C0	C2	7C	00061	CLRQ	PRIORITY_DESC	1783
00C8	C2	7C	00065	CLRQ	PRIVILEGE_DESC	1785
00D0	C2	7C	00069	CLRQ	PROCESS_DESC	1787
10	A2	7C	0006D	CLRQ	QUAD_CPU_TIME	1789
00D8	C2	7C	00070	CLRQ	QUEUE_DESC	1791
00E0	C2	7C	00074	CLRQ	QUEUE_TIME_DESC	1793
00E8	C2	7C	00078	CLRQ	REMOTE_ID_DESC	1795
00F0	C2	7C	0007C	CLRQ	START_TIME_DESC	1797
00F8	C2	7C	00080	CLRQ	STATUS_DESC	1799
0100	C2	7C	00084	CLRQ	STATUS-TEXT_DESC	1801
0108	C2	7C	00088	CLRQ	SUBTYPE_DESC	1803
0110	C2	7C	0008C	CLRQ	SYM_GET_DESC	1805
0120	C2	7C	00090	CLRQ	SYM_PAGE_DESC	1807
0118	C2	7C	00094	CLRQ	SYM_QIO_DESC	1809
0128	C2	7C	00098	CLRQ	TERMINAL_DESC	1811
0130	C2	7C	0009C	CLRQ	TYPE_DESC	1813
0138	C2	7C	000A0	CLRQ	TYPE-TEXT_DESC	1815
0158	C2	7C	000A4	CLRQ	UIC_DESC	1817
0140	C2	7C	000A8	CLRQ	UIC_BINARY_DESC	1819
0148	C2	7C	000AC	CLRQ	UIC_GROUP_DESC	1821
0150	C2	7C	000B0	CLRQ	UIC_MEMBER_DESC	1823
0160	C2	7C	000B4	CLRQ	USER_DATA_DESC	1825
0168	C2	7C	000B8	CLRQ	USER_DESC	1827
0170	C2	7C	000BC	CLRQ	VOLUMES_DESC	1829
20	A2	7C	000C0	CLRQ	WAIT_TIME	1831
0178	C2	7C	000C3	CLRQ	WORKING_SET_DESC	1833
B4	A2	7C	000C7	CLRQ	HOLD-BI0CNT	1836
BC	A2	7C	000CA	CLRQ	HOLD-ELAPSED	1838
C4	A2	7C	000CD	CLRQ	HOLD-EXECUTION	1840
CC	A2	7C	000D0	CLRQ	HOLD-GETCNT	1842
D4	A2	7C	000D3	CLRQ	HOLD-PAGCNT	1844
DC	A2	7C	000D6	CLRQ	HOLD-PAGEREADS	1846
E4	A2	7C	000D9	CLRQ	HOLD-CPUTIM	1848
EC	A2	D4	000DC	CLRL	HOLD-VOLUMES	1850

62	04	BC	03	0C	EF	000DF	EXTZ /	#12, #3, @BUFFER, VERSION	1852	
			50	62	D0	000E5	MOVL	VERSION, R0	1854	
				0A	12	000E8	BNEQ	1\$	1856	
				04	AC	DD 000EA	PUSHL	BUFFER	1857	
	0000V	CF		01	FB	000ED	CALLS	#1, PROCESS_V2_RECORD		
				10	11	000F2	BRB	3\$		
		02		50	D1	000F4	1\$:	CMPL	R0, #2	1858
				03	1B	000F7	BLEQU	2\$		
				0123	31	000F9	BRW	21\$		
				04	AC	DD 000FC	2\$:	PUSHL	BUFFER	1859
	0000V	CF		01	FB	000FF	CALLS	#1, PROCESS_V3_RECORD		
		01		50	EB	00104	3\$:	BLBS	STATUS, 4\$	
					04	00107	RET			
	03	02	A3		04	E0 00108	4\$:	BBS	#4, QUALIFIERS+2, 5\$	1864
				008D	31	0010D	BRW	17\$		
				38	A2	D5 00110	5\$:	TSTL	BUFFERED_IO_DESC	1866
					05	13 00113	BEQL	6\$		
	B4	A2		3C	B2	D0 00115	MOVL	@BUFFERED_IO_DESC+4, HOLD_BIOCNT	1868	
				40	A2	D5 0011A	6\$:	TSTL	CPU_TIME_DESC	1870
					05	13 0011D	BEQL	7\$		
	E4	A2		10	A2	7D 0011F	MOVQ	QUAD CPU TIME, HOLD_CPUTIM	1872	
				48	A2	D5 00124	7\$:	TSTL	DIRECT_IO_DESC	1875
					05	13 00127	BEQL	8\$		
	B8	A2		4C	B2	D0 00129	MOVL	@DIRECT_IO_DESC+4, HOLD_DIOCNT	1877	
	BC	A2		18	A2	7D 0012E	8\$:	MOVQ	ELAPSED_TIME, HOLD_ELAPSED	1879
				60	A2	D5 00133	TSTL	EXECUTION_DESC	1884	
					05	13 00136	BEQL	9\$		
	C4	A2		64	B2	D0 00138	MOVL	@EXECUTION_DESC+4, HOLD_EXECUTION	1886	
				00A8	C2	D5 0013D	9\$:	TSTL	PAGE_FAULTS_DESC	1888
					06	13 00141	BEQL	10\$		
	C8	A2		00AC	D2	D0 00143	MOVL	@PAGE_FAULTS_DESC+4, HOLD_PAGEFLTS	1890	
				00B0	C2	D5 00149	10\$:	TSTL	PAGE_FILE_DESC	1892
					06	13 0014D	BEQL	11\$		
	D8	A2		00B4	D2	D0 0014F	MOVL	@PAGE_FILE_DESC+4, HOLD_PGFLPEAK	1894	
				00B8	C2	D5 00155	11\$:	TSTL	PAGE_READS_DESC	1896
					06	13 00159	BEQL	12\$		
	DC	A2		00BC	D2	D0 0015B	MOVL	@PAGE_READS_DESC+4, HOLD_PAGEREADS	1898	
				0110	C2	D5 00161	12\$:	TSTL	SYM_GET_DESC	1900
					06	13 00165	BEQL	13\$		
	CC	A2		0114	D2	D0 00167	MOVL	@SYM_GET_DESC+4, HOLD_GETCNT	1902	
				0118	C2	D5 0016D	13\$:	TSTL	SYM_QIO_DESC	1904
					06	13 00171	BEQL	14\$		
	D0	A2		011C	D2	D0 00173	MOVL	@SYM_QIO_DESC+4, HOLD_QIOCNT	1906	
				0120	C2	D5 00179	14\$:	TSTL	SYM_PAGE_DESC	1908
					06	13 0017D	BEQL	15\$		
	D4	A2		0124	D2	D0 0017F	MOVL	@SYM_PAGE_DESC+4, HOLD_PAGCNT	1910	
				0170	C2	D5 00185	15\$:	TSTL	VOLUMES_DESC	1912
					06	13 00189	BEQL	16\$		
	EC	A2		0174	D2	D0 0018B	MOVL	@VOLUMES_DESC+4, HOLD_VOLUMES	1914	
				0178	C2	D5 00191	16\$:	TSTL	WORKING_SET_DESC	1916
					06	13 00195	BEQL	17\$		
	E0	A2		017C	D2	D0 00197	MOVL	@WORKING_SET_DESC+4, HOLD_WSPEAK	1918	
				02	A3	95 0019D	17\$:	TSTB	QUALIFIERS+2	1922
					05	19 001A0	BLSS	18\$		
				03	A3	95 001A2	TSTB	QUALIFIERS+3		
					16	18 001A5	BGEQ	19\$		
					7E	D4 001A7	18\$:	CLRL	-(SP)	1930

		040C	D2	DD	001A9	PUSHL	@DATE_DESC_ADDR		
		028C	C2	9F	001AD	PUSHAB	A_DATE_DESC		
			7E	D4	001B1	CLRL	-TSP)		
	00000000G	00	04	FB	001B3	CALLS	#4, LIB\$SYS_ASCTIM		
		64	50	E9	001BA	BLBC	STATUS, 22\$		
			03	A3	95 001BD	TSTB	QUALIFIERS+3	1933	
				59	18 001C0	BGEQ	20\$		
		040C	D2	DD	001C2	PUSHL	@DATE_DESC_ADDR	1944	
		02D0	C2	9F	001C6	PUSHAB	WEEKDAY		
	00000000G	00	02	FB	001CA	CALLS	#2, LIB\$DAY		
		4D	50	E9	001D1	BLBC	STATUS, 22\$		
7E	03	02D0	01	7A	001D4	EMUL	#1, WEEKDAY, #3, -(SP)	1946	
50	50		07	7B	001DB	EDIV	#7, (SP)+, R0, R0		
		02D0				MOVAB	48(R0), WEEKDAY		
			30	A0	9E 001E0	PUSHL	@DATE_DESC_ADDR	1955	
		040C	D2	DD	001E6	PUSHAB	NUMERIC_DATE		
	00000000G	00	C2	9F	001EA	CALLS	#2, SYS\$NUMTIM		
		29	02	FB	001EE	BLBC	STATUS, 22\$		
		7E	50	E9	001F5	BLBC	STATUS, 22\$	1963	
		7E	0298	C2	3C 001F8	MOVZWL	NUMERIC_DATE+4, -(SP)		
		7E	0296	C2	3C 001FD	MOVZWL	NUMERIC_DATE+2, -(SP)		
			0294	C2	3C 00202	MOVZWL	NUMERIC_DATE, -(SP)		
			02B0	C2	9F 00207	PUSHAB	S_DATE_DESC		
				7E	D4 0020B	CLRL	-TSP)		
		FDE7	CF	9F	0020D	PUSHAB	P.AAC		
	00000000G	00	06	FB	00211	CALLS	#6, LIB\$SYS_FAO		
		06	50	E9	00218	BLBC	STATUS, 22\$	1966	
		50	01	D0	0021B	MOVL	#1, R0		
				04	0021E	RET			
			50	D4	0021F	CLRL	R0	1968	
				04	00221	RET			

; Routine Size: 546 bytes, Routine Base: CODE + 03F0


```
1139 1969 1 ROUTINE PROCESS_V2_RECORD (BUFFER) =
1140 1970 1
1141 1971 1 ----
1142 1972 1
1143 1973 1 Functional description
1144 1974 1
1145 1975 1 This routine is called to analyze a record from the system
1146 1976 1 accounting file.
1147 1977 1
1148 1978 1 Input parameters
1149 1979 1
1150 1980 1 BUFFER = Address of a buffer containing the record to be detailed.
1151 1981 1 The record can be any type of record from the V2 format.
1152 1982 1 -- (ACC$W_MSGTYP can have any value)
1153 1983 1
1154 1984 1 Output parameters
1155 1985 1
1156 1986 1 Based on the contents of the record, values are established.
1157 1987 1 These values are used in later processing.
1158 1988 1 Any errors encountered are RETURNed immediately.
1159 1989 1 TRUE is returned on a normal exit.
1160 1990 1
1161 1991 1 ----
1162 1992 1
1163 1993 2 BEGIN
1164 1994 2
1165 1995 2 MAP
1166 1996 2 buffer: ref bblock; ! Define buffer block format
1167 1997 2
1168 1998 2 LOCAL
1169 1999 2 day_number, ! Temporary output of LIB$DAY
1170 2000 2 day_time; ! Temporary output of LIB$DAY
1171 2001 2
1172 2002 2
1173 2003 2 record_type = .buffer [acc$w_msgtyp];
1174 2004 2 SELECT ONEU .record_type of
1175 2005 2 SET
1176 2006 2 [acc$k_logtrm]: ! Login failure
1177 2007 2 BEGIN
1178 2008 2 type_desc [0] = 7;
1179 2009 2 type_desc [1] = uplit ('LOGFAIL');
1180 2010 2 type_text_desc [0] = 13;
1181 2011 2 type_text_desc [1] = uplit ('LOGIN FAILURE');
1182 2012 2 END;
1183 2013 2
1184 2014 2 [acc$k_insmg]: ! Arbitrary message
1185 2015 2 BEGIN
1186 2016 2 type_desc [0] = 4;
1187 2017 2 type_desc [1] = uplit ('USER');
1188 2018 2 type_text_desc [0] = 12;
1189 2019 2 type_text_desc [1] = uplit ('USER Message');
1190 2020 2 END;
1191 2021 2
1192 2022 2 [acc$k_inttrm]: ! Interactive process
1193 2023 2 BEGIN
1194 2024 2 type_desc [0] = 7;
1195 2025 2 type_desc [1] = uplit ('PROCESS');
```

```
1196 2026 3      subtype_desc [0]      = 11;
1197 2027 3      subtype_desc [1]      = uplit ('INTERACTIVE');
1198 2028 3      process_desc [0]       = .subtype_desc [0];
1199 2029 3      process_desc [1]       = .subtype_desc [1];
1200 2030 3      type_text_desc [0]     = 19;
1201 2031 3      type_text_desc [1]     = uplit ('Process Termination');
1202 2032 3      END;
1203 2033 3 [acc$k_prcrm]:      ! Non-interactive process
1204 2034 3      BEGIN
1205 2035 3      type_desc [0]           = 7;
1206 2036 3      type_desc [1]           = uplit ('PROCESS');
1207 2037 3      type_text_desc [0]      = 19;
1208 2038 3      type_text_desc [1]      = uplit ('Process Termination');
1209 2039 3      if .buffer [acc$l_owner] NEQ 0
1210 2040 3      then
1211 2041 3          BEGIN
1212 2042 3          subtype_desc [0]      = 10;
1213 2043 3          subtype_desc [1]      = uplit ('SUBPROCESS');
1214 2044 3          END
1215 2045 3      else
1216 2046 3          BEGIN
1217 2047 3          subtype_desc [0]      = 8;
1218 2048 3          subtype_desc [1]      = uplit ('DETACHED');
1219 2049 3          END;
1220 2050 3      process_desc [0]         = .subtype_desc [0];
1221 2051 3      process_desc [1]         = .subtype_desc [1];
1222 2052 3      END;
1223 2053 3 [acc$k_battrm]:      ! Batch job
1224 2054 3      BEGIN
1225 2055 3      type_desc [0]             = 7;
1226 2056 3      type_desc [1]             = uplit ('PROCESS');
1227 2057 3      subtype_desc [0]        = 5;
1228 2058 3      subtype_desc [1]        = uplit ('BATCH');
1229 2059 3      process_desc [0]        = .subtype_desc [0];
1230 2060 3      process_desc [1]        = .subtype_desc [1];
1231 2061 3      type_text_desc [0]      = 19;
1232 2062 3      type_text_desc [1]      = uplit ('Process Termination');
1233 2063 3      END;
1234 2064 3 [acc$k_prtjob]:      ! Print job
1235 2065 3      BEGIN
1236 2066 3      type_desc [0]             = 5;
1237 2067 3      type_desc [1]             = uplit ('PRINT');
1238 2068 3      type_text_desc [0]         = 21;
1239 2069 3      type_text_desc [1]         = uplit ('PRINT Job Termination');
1240 2070 3      END;
1241 2071 3 [otherwise]:      ! Unknown record type
1242 2072 3      BEGIN
1243 2073 3      type_desc [0]             = 7;
1244 2074 3      type_desc [1]             = uplit ('UNKNOWN');
1245 2075 3      type_text_desc [0]         = 19;
1246 2076 3      type_text_desc [1]         = uplit ('UNKNOWN Record Type');
1247 2077 3      END;
1248 2078 3
1249 2079 3
1250 2080 3
1251 2081 3
1252 2082 3
```

```

1253 2083 2 record_length      = .buffer [acc$w_msgsize];
1254 2084 2 status_desc [0]    = 4;
1255 2085 2 status_desc [1]    = buffer [acc$l_finalsts];
1256 2086 2 status_text_desc [0] = 80;
1257 2087 2 status_text_desc [1] = status_text_buffer;
1258 2088 2 translate_status (...status_desc [1], status_text_desc);
1259 2089 2 ident_desc [0]      = 4;
1260 2090 2 ident_desc [1]      = buffer [acc$l_pid];
1261 2091 2 if .buffer [acc$l_jobid] NEQ 0 then
1262 2092 2     BEGIN
1263 2093 2         entry_desc [0] = 4;
1264 2094 2         entry_desc [1] = buffer [acc$l_jobid];
1265 2095 2     END;
1266 2096 2 finish_time_desc [0] = 8;
1267 2097 2 finish_time_desc [1] = buffer [acc$q_termtime];
1268 2098 2 account_desc [0]     = acc$s_account;
1269 2099 2 account_desc [1]     = buffer [acc$t_account];
1270 2100 2 strip_trail        = (account_desc);
1271 2101 2 user_desc [0]       = acc$s_username;
1272 2102 2 user_desc [1]       = buffer [acc$t_username];
1273 2103 2 strip_trail        = (user_desc);
1274 2104 2
1275 2105 2 SELECTU .record_type of
1276 2106 2     SET
1277 2107 2     [acc$k_inttrm, acc$k_prctrm, acc$k_battrm, acc$k_logtrm]:
1278 2108 2     BEGIN
1279 2109 2
1280 2110 2         BUILTIN emul;
1281 2111 2
1282 2112 2         cpu_time_desc [0] = 4;
1283 2113 2         cpu_time_desc [1] = buffer [acc$l_cputim];
1284 2114 2         emul (%ref(100000), buffer [acc$l_cputim], %ref(0),
1285 2115 2             quad_cpu_time [0]);
1286 2116 2         page_faults_desc [0] = 4;
1287 2117 2         page_faults_desc [1] = buffer [acc$l_pageflts];
1288 2118 2         page_file_desc [0]   = 4;
1289 2119 2         page_file_desc [1]   = buffer [acc$l_pgflpeak];
1290 2120 2         working_set_desc [0] = 4;
1291 2121 2         working_set_desc [1] = buffer [acc$l_wspeak];
1292 2122 2         buffered_io_desc [0] = 4;
1293 2123 2         buffered_io_desc [1] = buffer [acc$l_biocnt];
1294 2124 2         direct_io_desc [0]   = 4;
1295 2125 2         direct_io_desc [1]   = buffer [acc$l_diocnt];
1296 2126 2         volumes_desc [0]     = 4;
1297 2127 2         volumes_desc [1]     = buffer [acc$l_volumes];
1298 2128 2         start_time_desc [0]  = 8;
1299 2129 2         start_time_desc [1]  = buffer [acc$q_login];
1300 2130 2         perform (lib$subx (
1301 2131 2             .finish_time_desc [1],
1302 2132 2             .start_time_desc [1],
1303 2133 2             elapsed_time: %ref(2)));
1304 2134 2         elapsed_desc [0]     = 8;
1305 2135 2         owner_desc [0]       = 4;
1306 2136 2         owner_desc [1]       = buffer [acc$l_owner];
1307 2137 2     END;
1308 2138 2
1309 2139 2     [acc$k_battrm]:
```

```
1310 2140 BEGIN
1311 2141 job_desc [0] = acc$s_job_name;
1312 2142 job_desc [1] = buffer [acc$t_job_name];
1313 2143 strip_trail (job_desc);
1314 2144 queue_desc [0] = .(buffer [acc$t_job_que])<0,8>;
1315 2145 queue_desc [1] = buffer [acc$t_job_que] + 1;
1316 2146 END;
1317 2147
1318 2148 [acc$k_prtinb]:
1319 2149 BEGIN
1320 2150
1321 2151 BUILTIN emul;
1322 2152
1323 2153 cpu_time_desc [0] = 4;
1324 2154 cpu_time_desc [1] = buffer [acc$l_cputim];
1325 2155 emul (%ref(100000), buffer [acc$l_cputim], %ref(0),
1326 2156 quad_cpu_time [0]);
1327 2157 sym_page_desc [0] = 4;
1328 2158 sym_page_desc [1] = buffer [acc$l_pagcnt];
1329 2159 sym_qio_desc [0] = 4;
1330 2160 sym_qio_desc [1] = buffer [acc$l_qiocnt];
1331 2161 sym_get_desc [0] = 4;
1332 2162 sym_get_desc [1] = buffer [acc$l_getcnt];
1333 2163 queue_time_desc [0] = 8;
1334 2164 queue_time_desc [1] = buffer [acc$q_quetime];
1335 2165 perform (lib$subx (
1336 2166 .finish_time_desc [1],
1337 2167 .queue_time_desc [1],
1338 2168 elapsed_time, %ref(2)));
1339 2169 elapsed_desc [0] = 8;
1340 2170 job_desc [0] = acc$s_prt_name;
1341 2171 job_desc [1] = buffer [acc$t_prt_name];
1342 2172 strip_trail (job_desc);
1343 2173 queue_desc [0] = .(buffer [acc$t_prt_que])<0,8>;
1344 2174 queue_desc [1] = buffer [acc$t_prt_que] + 1;
1345 2175 END;
1346 2176
1347 2177 [acc$k_insmg]:
1348 2178 BEGIN
1349 2179 user_data_desc [0] = .(buffer [acc$t_user_data])<0,8>;
1350 2180 user_data_desc [1] = buffer [acc$t_user_data] + 1;
1351 2181 strip_trail (user_data_desc);
1352 2182 END;
1353 2183 TES;
1354 2184
1355 2185
1356 2186 return true;
1357 2187
1358 2188 1 END;
```

```
00 00 45 52 55 4C 49 00 4C 49 41 46 47 4F 4C 00612 .BLKB 2
00 00 45 52 55 4C 49 41 46 20 4E 49 47 4F 4C 00614 P.AAE: .ASCII \LOGFAIL\<0>
00 00 45 52 55 4C 49 41 46 20 4E 49 47 4F 4C 0061C P.AAF: .ASCII \LOGIN FAILURE\<0><0><0>
00 00 45 52 55 4C 49 41 46 20 4E 49 47 4F 4C 0062B
00 00 45 52 55 4C 49 41 46 20 4E 49 47 4F 4C 0062C P.AAG: .ASCII \USER\
```

00FC 00000 PROCESS_V2 RECORD:				.WORD Save R2,R3,R4,R5,R6,R7		1969
	57	00000000G	00 9E 00002	MOVAB	LIBSSUBX, R7	...
	56	00000000G	00 9E 00009	MOVAB	STRIP_TRAIL, R6	...
	55	FF0C	CF 9E 00010	MOVAB	P.AAE, R5	...
	54	00000000'	EF 9E 00015	MOVAB	TYPE_DESC, R4	...
	5E		04 C2 0001C	SUBL2	#4, SP	...
FECC	52	04	AC D0 0001F	MOVL	BUFFER, R2	2003
	C4		62 3C 00023	MOVZWL	(R2), RECORD_TYPE	...
	50	FECC	C4 D0 00028	MOVL	RECORD_TYPE, R0	2004
	04		50 D1 0002D	CMPL	R0, #4	2006
			12 12 00030	BNEQ	1\$	...
	64		07 D0 00032	MOVL	#7, TYPE_DESC	2008
04	A4		65 9E 00035	MOVAB	P.AAE, TYPE_DESC+4	2009
08	A4		0D D0 00039	MOVL	#13, TYPE_TEXT_DESC	2010
0C	A4	08	A5 9E 0003D	MOVAB	P.AAF, TYPE_TEXT_DESC+4	2011
			72 11 00042	BRB	6\$	2004
	11		50 D1 00044	CMPL	R0, #17	2014
			13 12 00047	BNEQ	2\$	...
	64		04 D0 00049	MOVL	#4, TYPE_DESC	2016
04	A4	18	A5 9E 0004C	MOVAB	P.AAG, TYPE_DESC+4	2017
08	A4		0C D0 00051	MOVL	#12, TYPE_TEXT_DESC	2018
0C	A4	1C	A5 9E 00055	MOVAB	P.AAH, TYPE_TEXT_DESC+4	2019
			5A 11 0005A	BRB	6\$	2004
	03		50 D1 0005C	CMPL	R0, #3	2022
			21 12 0005F	BNEQ	3\$	...
	64		07 D0 00061	MOVL	#7, TYPE_DESC	2024
04	A4	28	A5 9E 00064	MOVAB	P.AAI, TYPE_DESC+4	2025
D8	A4		0B D0 00069	MOVL	#11, SUBTYPE_DESC	2026
DC	A4	30	A5 9E 0006D	MOVAB	P.AAJ, SUBTYPE_DESC+4	2027
A0	A4	D8	A4 7D 00072	MOVQ	SUBTYPE_DESC, PROCESS_DESC	2028
08	A4		13 D0 00077	MOVL	#19, TYPE_TEXT_DESC	2030
0C	A4	3C	A5 9E 0007B	MOVAB	P.AAK, TYPE_TEXT_DESC+4	2031
			77 11 00080	BRB	9\$	2004

	01		50	D1	00082	3\$:	CMPL	R0, #1	2033
			31	12	00085		BNEQ	7\$	
	64		07	D0	00087		MOVL	#7, TYPE_DESC	2035
04	A4	50	A5	9E	0008A		MOVAB	P.AAL, TYPE_DESC+4	2036
08	A4		13	D0	0008F		MOVL	#19, TYPE_TEXT_DESC	2037
0C	A4	58	A5	9E	00093		MOVAB	P.AAM, TYPE_TEXT_DESC+4	2038
		50	A2	D5	00098		TSTL	80(R2)	2039
			08	13	0009B		BEQL	4\$	
D8	A4		0A	D0	0009D		MOVL	#10, SUBTYPE_DESC	2042
DC	A4	6C	A5	9E	000A1		MOVAB	P.AAN, SUBTYPE_DESC+4	2043
			09	11	000A6		BRB	5\$	2039
D8	A4		08	D0	000AB	4\$:	MOVL	#8, SUBTYPE_DESC	2047
DC	A4	78	A5	9E	000AC		MOVAB	P.AAO, SUBTYPE_DESC+4	2048
A0	A4	D8	A4	7D	000B1	5\$:	MOVQ	SUBTYPE_DESC, PROCESS_DESC	2050
			56	11	000B6	6\$:	BRB	11\$	2004
	02		50	D1	000B8	7\$:	CMPL	R0, #2	2054
			24	12	000BB		BNEQ	8\$	
	64		07	D0	000BD		MOVL	#7, TYPE_DESC	2056
04	A4	0080	C5	9E	000C0		MOVAB	P.AAP, TYPE_DESC+4	2057
D8	A4		05	D0	000C6		MOVL	#5, SUBTYPE_DESC	2058
DC	A4	0088	C5	9E	000CA		MOVAB	P.AAQ, SUBTYPE_DESC+4	2059
A0	A4	D8	A4	7D	000D0		MOVQ	SUBTYPE_DESC, PROCESS_DESC	2060
08	A4		13	D0	000D5		MOVL	#19, TYPE_TEXT_DESC	2062
0C	A4	0090	C5	9E	000D9		MOVAB	P.AAR, TYPE_TEXT_DESC+4	2063
			2D	11	000DF		BRB	11\$	2004
	10		50	D1	000E1	8\$:	CMPL	R0, #16	2066
			15	12	000E4		BNEQ	10\$	
	64		05	D0	000E6		MOVL	#5, TYPE_DESC	2068
04	A4	00A4	C5	9E	000E9		MOVAB	P.AAS, TYPE_DESC+4	2069
08	A4		15	D0	000EF		MOVL	#21, TYPE_TEXT_DESC	2070
0C	A4	00AC	C5	9E	000F3		MOVAB	P.AAT, TYPE_TEXT_DESC+4	2071
			13	11	000F9	9\$:	BRB	11\$	2004
	64		07	D0	000FB	10\$:	MOVL	#7, TYPE_DESC	2076
04	A4	00C4	C5	9E	000FE		MOVAB	P.AAU, TYPE_DESC+4	2077
08	A4		13	D0	00104		MOVL	#19, TYPE_TEXT_DESC	2078
0C	A4	00CC	C5	9E	00108		MOVAB	P.AAV, TYPE_TEXT_DESC+4	2079
FEC4	C4	02	A2	3C	0010E	11\$:	MOVZWL	2(R2), RECORD_LENGTH	2083
C8	A4		04	D0	00114		MOVL	#4, STATUS_DESC	2084
CC	A4	04	A2	9E	00118		MOVAB	4(R2), STATUS_DESC+4	2085
D0	A4	50	8F	9A	0011D		MOVZBL	#80, STATUS_TEXT_DESC	2086
D4	A4	00B0	C4	9E	00122		MOVAB	STATUS_TEXT_BUFFER, STATUS_TEXT_DESC+4	2087
		D0	A4	9F	00128		PUSHAB	STATUS_TEXT_DESC	2088
		CC	B4	DD	0012B		PUSHL	@STATUS_DESC+4	
00000000G	00		02	FB	0012E		CALLS	#2, TRANSLATE_STATUS	
FF48	C4		04	D0	00135		MOVL	#4, IDENT_DESC	2089
FF4C	C4	08	A2	9E	0013A		MOVAB	8(R2), IDENT_DESC+4	2090
		0C	A2	D5	00140		TSTL	12(R2)	2091
			08	13	00143		BEQL	12\$	
FF20	C4		04	D0	00145		MOVL	#4, ENTRY_DESC	2093
FF24	C4	0C	A2	9E	0014A		MOVAB	12(R2), ENTRY_DESC+4	2094
FF40	C4		08	D0	00150	12\$:	MOVL	#8, FINISH_TIME_DESC	2096
FF44	C4	10	A2	9E	00155		MOVAB	16(R2), FINISH_TIME_DESC+4	2097
FEF8	C4		08	D0	0015B		MOVL	#8, ACCOUNT_DESC	2098
FEFC	C4	18	A2	9E	00160		MOVAB	24(R2), ACCOUNT_DESC+4	2099
		FEF8	C4	9F	00166		PUSHAB	ACCOUNT_DESC	2100
	66		01	FB	0016A		CALLS	#1, STRIP_TRAIL	
38	A4		0C	D0	0016D		MOVL	#12, USER_DESC	2101

	3C	A4	20	A2	9E	00171	MOVAB	32(R2), USER_DESC+4	2102
			38	A4	9F	00176	PUSHAB	USER_DESC	2103
	66			01	FB	00179	CALLS	#1, STRIP TRAIL	
	53	FECC		C4	D0	0017C	MOVL	RECORD_TYPE, R3	2105
				03	12	00181	BNEQ	14\$	2107
				0088	31	00183	BRW	16\$	
	04			53	D1	00186	CMPL	R3, #4	
				F8	1A	00189	BGTRU	13\$	
	FF10	C4		04	D0	0018B	MOVL	#4, CPU_TIME_DESC	2112
	FF14	C4	2C	A2	9E	00190	MOVAB	44(R2), CPU_TIME_DESC+4	2113
FEE0	2C	A2	000186A0	8F	7A	00196	EMUL	#100000, 44(R2), #0, QUAD_CPU_TIME	2115
	FF78	C4		04	D0	001A2	MOVL	#4, PAGE_FAULTS_DESC	2116
	FF7C	C4	30	A2	9E	001A7	MOVAB	48(R2), PAGE_FAULTS_DESC+4	2117
	80	A4		04	D0	001AD	MOVL	#4, PAGE_FILE_DESC	2118
	84	A4	34	A2	9E	001B1	MOVAB	52(R2), PAGE_FILE_DESC+4	2119
	48	A4		04	D0	001B6	MOVL	#4, WORKING_SET_DESC	2120
	4C	A4	38	A2	9E	001BA	MOVAB	56(R2), WORKING_SET_DESC+4	2121
	FF08	C4		04	D0	001BF	MOVL	#4, BUFFERED_IO_DESC	2122
	FF0C	C4	3C	A2	9E	001C4	MOVAB	60(R2), BUFFERED_IO_DESC+4	2123
	FF18	C4		04	D0	001CA	MOVL	#4, DIRECT_IO_DESC	2124
	FF1C	C4	40	A2	9E	001CF	MOVAB	64(R2), DIRECT_IO_DESC+4	2125
	40	A4		04	D0	001D5	MOVL	#4, VOLUMES_DESC	2126
	44	A4	44	A2	9E	001D9	MOVAB	68(R2), VOLUMES_DESC+4	2127
	C0	A4		08	D0	001DE	MOVL	#8, START_TIME_DESC	2128
	C4	A4	48	A2	9E	001E2	MOVAB	72(R2), START_TIME_DESC+4	2129
		6E		02	D0	001E7	MOVL	#2, (SP)	2133
				5E	DD	001EA	PUSHL	SP	
		FEE8		C4	9F	001EC	PUSHAB	ELAPSED_TIME	
		C4		A4	DD	001F0	PUSHL	START_TIME_DESC+4	
		FF44		C4	DD	001F3	PUSHL	FINISH_TIME_DESC+4	
				04	FB	001F7	CALLS	#4, LIB\$SUBX	
	67			50	E8	001FA	BLBS	STATUS, 15\$	
	C1			04	001FD	RET			
	FF28	C4		08	D0	001FE	MOVL	#8, ELAPSED_DESC	2134
	FF70	C4		04	D0	00203	MOVL	#4, OWNER_DESC	2135
	FF74	C4	50	A2	9E	00208	MOVAB	80(R2), OWNER_DESC+4	2136
		02		53	D1	0020E	CMPL	R3, #2	2139
				1C	12	00211	BNEQ	17\$	
	FF60	C4		08	D0	00213	MOVL	#8, JOB_DESC	2141
	FF64	C4	54	A2	9E	00218	MOVAB	84(R2), JOB_DESC+4	2142
			FF60	C4	9F	0021E	PUSHAB	JOB_DESC	2143
		66		01	FB	00222	CALLS	#1, STRIP TRAIL	
	A8	A4	5C	A2	9A	00225	MOVZBL	92(R2), QUEUE_DESC	2144
	AC	A4	5D	A2	9E	0022A	MOVAB	93(R2), QUEUE_DESC+4	2145
		10		53	D1	0022F	CMPL	R3, #16	2148
				72	12	00232	BNEQ	18\$	
	FF10	C4		04	D0	00234	MOVL	#4, CPU_TIME_DESC	2153
	FF14	C4	2C	A2	9E	00239	MOVAB	44(R2), CPU_TIME_DESC+4	2154
FEE0	2C	A2	000186A0	8F	7A	0023F	EMUL	#100000, 44(R2), #0, QUAD_CPU_TIME	2156
	F0	A4		04	D0	0024B	MOVL	#4, SYM_PAGE_DESC	2157
	F4	A4	30	A2	9E	0024F	MOVAB	48(R2), SYM_PAGE_DESC+4	2158
	E8	A4		04	D0	00254	MOVL	#4, SYM_QIO_DESC	2159
	EC	A4	34	A2	9E	00258	MOVAB	52(R2), SYM_QIO_DESC+4	2160
	E0	A4		04	D0	0025D	MOVL	#4, SYM_GET_DESC	2161
	E4	A4	38	A2	9E	00261	MOVAB	56(R2), SYM_GET_DESC+4	2162
	B0	A4		08	D0	00266	MOVL	#8, QUEUE_TIME_DESC	2163
	B4	A4	3C	A2	9E	0026A	MOVAB	60(R2), QUEUE_TIME_DESC+4	2164

ACC
V04-000

F 7
15-Sep-1984 23:36:11 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 11:51:58 DISK\$VMSMASTER:[ACC.SRC]ACC.B32;1

Page 46
(9)

	6E		02	DD	0026F	MOVL	#2, (SP)	:	2168
			5E	DD	00272	PUSHL	SP	:	
		FEE8	C4	9F	00274	PUSHAB	ELAPSED TIME	:	
		B4	A4	DD	00278	PUSHL	QUEUE TIME_DESC+4	:	
		FF44	C4	DD	0027B	PUSHL	FINISH TIME_DESC+4	:	
	67		04	FB	0027F	CALLS	#4, LIB\$SUBR	:	
	39		50	E9	00282	BLBC	STATUS, 20\$	:	
FF28	C4		08	DD	00285	MOVL	#8, ELAPSED_DESC	:	2169
FF60	C4		08	DD	0028A	MOVL	#8, JOB_DESC	:	2170
FF64	C4	44	A2	9E	0028F	MOVAB	68(R2), JOB_DESC+4	:	2171
		FF60	C4	9F	00295	PUSHAB	JOB_DESC	:	2172
	66		01	FB	00299	CALLS	#1, STRIP_TRAIL	:	
A8	A4	4C	A2	9A	0029C	MOVZBL	76(R2), QUEUE_DESC	:	2173
AC	A4	4D	A2	9E	002A1	MOVAB	77(R2), QUEUE_DESC+4	:	2174
	11		53	D1	002A6	CMPL	R3, #17	:	2177
			10	12	002A9	BNEQ	19\$	:	
30	A4	2C	A2	9A	002AB	MOVZBL	44(R2), USER_DATA_DESC	:	2179
34	A4	2D	A2	9E	002B0	MOVAB	45(R2), USER_DATA_DESC+4	:	2180
		30	A4	9F	002B5	PUSHAB	USER_DATA_DESC	:	2181
	66		01	FB	002B8	CALLS	#1, STRIP_TRAIL	:	
	50		01	DD	002BB	MOVL	#1, R0	:	2186
			04	002BE	20\$:	RET		:	2188

; Routine Size: 703 bytes, Routine Base: CODE + 06F4


```
: 1360 2189 1 ROUTINE PROCESS_V3_RECORD (BUFFER) =
: 1361 2190 1
: 1362 2191 1 ----
: 1363 2192 1
: 1364 2193 1 Functional description
: 1365 2194 1
: 1366 2195 1 This routine is called to analyze a record from the system
: 1367 2196 1 accounting file.
: 1368 2197 1
: 1369 2198 1 Input parameters
: 1370 2199 1
: 1371 2200 1 BUFFER = Address of a buffer containing the record to be detailed.
: 1372 2201 1 The record can be any type of record from the V3 format.
: 1373 2202 1 -- (ACRSW_TYPE can have any value)
: 1374 2203 1
: 1375 2204 1 Output parameters
: 1376 2205 1
: 1377 2206 1 Based on the contents of the record, values are established.
: 1378 2207 1 These values are used in later processing.
: 1379 2208 1 Any errors encountered are RETURNed immediately.
: 1380 2209 1 TRUE is returned on a normal exit.
: 1381 2210 1
: 1382 2211 1 ----
: 1383 2212 1
: 1384 2213 2 BEGIN
: 1385 2214 2
: 1386 2215 2 MAP
: 1387 2216 2 buffer: ref bblock; ! Define buffer block format
: 1388 2217 2
: 1389 2218 2 LABEL
: 1390 2219 2 pkt;
: 1391 2220 2
: 1392 2221 2 OWN
: 1393 2222 2 packet;
```

```
1395 2223 2 -----
1396 2224 2
1397 2225 2 Record header
1398 2226 2 -----
1399 2227 2
1400 2228 2
1401 2229 2 Record type and subtype
1402 2230 2
1403 2231 2 if .buffer [acr$v_packet]
1404 2232 2 then return false;
1405 2233 2 record_type = .buffer [acr$v_type];
1406 2234 2 record_subtype = .buffer [acr$v_subtype];
1407 2235 2 customer = .buffer [acr$v_customer];
1408 2236 2 if NOT .customer
1409 2237 2 then
1410 2238 2 BEGIN
1411 2239 2 SELECTU .record_type of
1412 2240 2 SET
1413 2241 2 [acr$k_prdel, acr$k_prpur, acr$k_imgdel, acr$k_imgdel]:
1414 2242 2 BEGIN
1415 2243 2 SELECTONEU .record_subtype of
1416 2244 2 SET
1417 2245 2 [acr$k_interactive]:
1418 2246 2 BEGIN
1419 2247 2 process_desc [0] = 11;
1420 2248 2 process_desc [1] = uplit ('INTERACTIVE');
1421 2249 2 END;
1422 2250 2 [acr$k_subprocess]:
1423 2251 2 BEGIN
1424 2252 2 process_desc [0] = 10;
1425 2253 2 process_desc [1] = uplit ('SUBPROCESS');
1426 2254 2 END;
1427 2255 2 [acr$k_detached]:
1428 2256 2 BEGIN
1429 2257 2 process_desc [0] = 8;
1430 2258 2 process_desc [1] = uplit ('DETACHED');
1431 2259 2 END;
1432 2260 2 [acr$k_batch]:
1433 2261 2 BEGIN
1434 2262 2 process_desc [0] = 5;
1435 2263 2 process_desc [1] = uplit ('BATCH');
1436 2264 2 END;
1437 2265 2 [acr$k_network]:
1438 2266 2 BEGIN
1439 2267 2 process_desc [0] = 7;
1440 2268 2 process_desc [1] = uplit ('NETWORK');
1441 2269 2 END;
1442 2270 2 TES;
1443 2271 2 END;
1444 2272 2 [acr$k_prdel, acr$k_prpur]:
1445 2273 2 BEGIN
1446 2274 2 type_desc [0] = 7;
1447 2275 2 type_desc [1] = uplit ('PROCESS');
1448 2276 2 type_text_desc [0] = 19;
1449 2277 2 type_text_desc [1] = uplit ('Process Termination');
1450 2278 2 subtype_desc [0] = .process_desc [0];
1451 2279 2 subtype_desc [1] = .process_desc [1];
```

1452	2280	3	END;
1453	2281	3	[acr\$sk_imgdel, acr\$sk_imgpur]:
1454	2282	4	BEGIN
1455	2283	4	type_desc [0] = 5;
1456	2284	4	type_desc [1] = uplit ('IMAGE');
1457	2285	4	type_text_desc [0] = 17;
1458	2286	4	type_text_desc [1] = uplit ('Image Termination');
1459	2287	3	END;
1460	2288	3	[acr\$sk_sysinit]:
1461	2289	4	BEGIN
1462	2290	4	type_desc [0] = 7;
1463	2291	4	type_desc [1] = uplit ('SYSINIT');
1464	2292	4	type_text_desc [0] = 21;
1465	2293	4	type_text_desc [1] = uplit ('SYSTEM INITIALIZATION');
1466	2294	3	END;
1467	2295	3	[acr\$sk_settime]:
1468	2296	4	BEGIN
1469	2297	4	type_desc [0] = 7;
1470	2298	4	type_desc [1] = uplit ('SETTIME');
1471	2299	4	type_text_desc [0] = 18;
1472	2300	4	type_text_desc [1] = uplit ('System Time Change');
1473	2301	3	END;
1474	2302	3	[acr\$sk_logfail]:
1475	2303	4	BEGIN
1476	2304	4	type_desc [0] = 7;
1477	2305	4	type_desc [1] = uplit ('LOGFAIL');
1478	2306	4	type_text_desc [0] = 13;
1479	2307	4	type_text_desc [1] = uplit ('LOGIN FAILURE');
1480	2308	3	END;
1481	2309	3	[acr\$sk_print]:
1482	2310	4	BEGIN
1483	2311	4	type_desc [0] = 5;
1484	2312	4	type_desc [1] = uplit ('PRINT');
1485	2313	4	type_text_desc [0] = 21;
1486	2314	4	type_text_desc [1] = uplit ('PRINT Job Termination');
1487	2315	3	END;
1488	2316	3	[acr\$sk_user]:
1489	2317	4	BEGIN
1490	2318	4	type_desc [0] = 4;
1491	2319	4	type_desc [1] = uplit ('USER');
1492	2320	4	type_text_desc [0] = 12;
1493	2321	4	type_text_desc [1] = uplit ('USER Message');
1494	2322	3	END;
1495	2323	3	[acr\$sk_enable]:
1496	2324	4	BEGIN
1497	2325	4	type_desc [0] = 6;
1498	2326	4	type_desc [1] = uplit ('ENABLE');
1499	2327	4	type_text_desc [0] = 17;
1500	2328	4	type_text_desc [1] = uplit ('Enable Accounting');
1501	2329	3	END;
1502	2330	3	[acr\$sk_disable]:
1503	2331	4	BEGIN
1504	2332	4	type_desc [0] = 7;
1505	2333	4	type_desc [1] = uplit ('DISABLE');
1506	2334	4	type_text_desc [0] = 18;
1507	2335	4	type_text_desc [1] = uplit ('Disable Accounting');
1508	2336	3	END;

```
1509 2337 3 [acr$altacm]:
1510 2338 4 BEGIN
1511 2339 4 type_desc [0] = 6;
1512 2340 4 type_desc [1] = uplit ('ALTACM');
1513 2341 4 type_text_desc [0] = 28;
1514 2342 4 type_text_desc [1] = uplit ('Alternate Accounting Manager');
1515 2343 3 END;
1516 2344 3 [acr$file bl]:
1517 2345 4 BEGIN
1518 2346 4 type_desc [0] = 4;
1519 2347 4 type_desc [1] = uplit ('FILE_BL');
1520 2348 4 type_text_desc [0] = 32;
1521 2349 4 type_text_desc [1] = uplit
1522 2350 4 ('ACCOUNTING FILE backward pointer');
1523 2351 3 END;
1524 2352 3 [acr$file fl]:
1525 2353 4 BEGIN
1526 2354 4 type_desc [0] = 4;
1527 2355 4 type_desc [1] = uplit ('FILE_FL');
1528 2356 4 type_text_desc [0] = 31;
1529 2357 4 type_text_desc [1] = uplit
1530 2358 4 ('ACCOUNTING FILE forward pointer');
1531 2359 3 END;
1532 2360 3 [OTHERWISE]:
1533 2361 4 BEGIN
1534 2362 4 type_desc [0] = 7;
1535 2363 4 type_desc [1] = uplit ('UNKNOWN');
1536 2364 4 type_text_desc [0] = 19;
1537 2365 4 type_text_desc [1] = uplit ('UNKNOWN Record Type');
1538 2366 3 END;
1539 2367 3 TES;
1540 2368 3 END
1541 2369 2 else
1542 2370 3 BEGIN
1543 2371 3 type_desc [0] = 8;
1544 2372 3 type_desc [1] = uplit ('CUSTOMER');
1545 2373 3 type_text_desc [0] = 20;
1546 2374 3 type_text_desc [1] = uplit ('CUSTOMER Record Type');
1547 2375 2 END;
1548 2376 2
1549 2377 2 Record length
1550 2378 2
1551 2379 2 record_length = .buffer [acr$w_length];
1552 2380 2
1553 2381 2 Record creation time
1554 2382 2
1555 2383 2 if .(buffer [acr$q_systime])<0,32> NEQ 0 OR
1556 2384 2 .(buffer [acr$q_systime] + 4)<0,32> NEQ 0
1557 2385 2 then
1558 2386 3 BEGIN
1559 2387 3 finish_time_desc [0] = 8;
1560 2388 3 finish_time_desc [1] = buffer [acr$q_systime];
1561 2389 2 END;
```

```
1563 2390 2 -----
1564 2391 2
1565 2392 2 Packet processing
1566 2393 2 -----
1567 2394 2
1568 2395 2
1569 2396 2 Find and detail all known packets
1570 2397 2
1571 2398 2 packet = .buffer + $BYTEOFFSET(acr$k_hdrlen);
1572 2399 2 while .packet LSSU .buffer + .record_length do
1573 2400 2 BEGIN
1574 2401 2 MAP
1575 2402 2 packet: ref bblock;
1576 2403 2 PKT: BEGIN
1577 2404 2 if (.packet [acr$v_packet] EQL 0) then return false;
1578 2405 2 if (.packet [acr$v_customer] NEQ 0) then leave pkt;
1579 2406 2 SELECTONEU .packet [acr$v_type] of
1580 2407 2 SET
1581 2408 2 [acr$k_id]:
1582 2409 2 ID packet
1583 2410 2 BEGIN
1584 2411 2 BIND
1585 2412 2 id = .packet: bblock;
1586 2413 2
1587 2414 2 BEGIN
1588 2415 2 ident_desc [0] = 4;
1589 2416 2 ident_desc [1] = id [acr$l_pid];
1590 2417 2 END;
1591 2418 2
1592 2419 2 if (.id [acr$l_owner] NEQ 0) then
1593 2420 2 BEGIN
1594 2421 2 owner_desc [0] = 4;
1595 2422 2 owner_desc [1] = id [acr$l_owner];
1596 2423 2 END;
1597 2424 2
1598 2425 2 BEGIN
1599 2426 2 uic_binary_desc [0] = 4;
1600 2427 2 uic_binary_desc [1] = id [acr$l_uic];
1601 2428 2 uic_group_desc [0] = 2;
1602 2429 2 uic_group_desc [1] = id [acr$w_grp];
1603 2430 2 uic_member_desc [0] = 2;
1604 2431 2 uic_member_desc [1] = id [acr$w_mem];
1605 2432 2
1606 2433 2 If /SUMMARY=UIC then do UIC conversion. For
1607 2434 2 /BRIEF, /BINARY, or a /SUMMARY but not by UIC, we
1608 2435 2 won't need it, so avoid it. For /FULL, do it in
1609 2436 2 FORMAT\SHOW_FULL.
1610 2437 2
1611 2438 2 if .SUMMARY_UIC
1612 2439 2 then
1613 2440 2 begin
1614 2441 2 uic_desc [0] = 67;
1615 2442 2 uic_desc [1] = uic_buf;
1616 2443 2 perform (lib$sys_fao (ad('!%I'),
1617 2444 2
1618 2445 2
1619 P 2446 7
```

1620	P	2447	7
1621		2448	7
1622		2449	7
1623		2450	6
1624		2451	5
1625		2452	5
1626		2453	6
1627		2454	6
1628		2455	6
1629		2456	5
1630		2457	5
1631		2458	6
1632		2459	6
1633		2460	6
1634		2461	5
1635		2462	5
1636		2463	5
1637		2464	6
1638		2465	6
1639		2466	6
1640		2467	6
1641		2468	6
1642		2469	5
1643		2470	5
1644		2471	6
1645		2472	6
1646		2473	6
1647		2474	6
1648		2475	6
1649		2476	5
1650		2477	5
1651		2478	6
1652		2479	6
1653		2480	6
1654		2481	6
1655		2482	6
1656		2483	5
1657		2484	5
1658		2485	6
1659		2486	6
1660		2487	6
1661		2488	6
1662		2489	6
1663		2490	5
1664		2491	5
1665		2492	6
1666		2493	6
1667		2494	6
1668		2495	6
1669		2496	6
1670		2497	5
1671		2498	5
1672		2499	6
1673		2500	6
1674		2501	6
1675		2502	5
1676		2503	5

```
uic_desc [0],
uic_desc [0],
.id [acr$l_uic]));

end;
END;

BEGIN
privilege_desc [0] = 8;
privilege_desc [1] = id [acr$q_priv];
END;

BEGIN
priority_desc [0] = 1;
priority_desc [1] = id [acr$b_pri];
END;

if (.id [acr$w_username] NEQ 0) then
BEGIN
user_desc [0] =
.(id + .id [acr$w_username])<0,8>;
user_desc [1] =
id + .id [acr$w_username] + 1;
END;
if (.id [acr$w_account] NEQ 0) then
BEGIN
account_desc [0] =
.(id + .id [acr$w_account])<0,8>;
account_desc [1] =
id + .id [acr$w_account] + 1;
END;
if (.id [acr$w_nodename] NEQ 0) then
BEGIN
node_desc [0] =
.(id + .id [acr$w_nodename])<0,8>;
node_desc [1] =
id + .id [acr$w_nodename] + 1;
END;
if (.id [acr$w_terminal] NEQ 0) then
BEGIN
terminal_desc [0] =
.(id + .id [acr$w_terminal])<0,8>;
terminal_desc [1] =
id + .id [acr$w_terminal] + 1;
END;
if (.id [acr$w_jobname] NEQ 0) then
BEGIN
job_desc [0] =
.(id + .id [acr$w_jobname])<0,8>;
job_desc [1] =
id + .id [acr$w_jobname] + 1;
END;
if (.id [acr$l_jobid] NEQ 0) then
BEGIN
entry_desc [0] = 4;
entry_desc [1] = id [acr$l_jobid];
END;
if (.id [acr$w_queue] NEQ 0) then
```

1677 2504 6
1678 2505 6
1679 2506 6
1680 2507 6
1681 2508 6
1682 2509 5
1683 2510 5
1684 2511 6
1685 2512 6
1686 2513 7
1687 2514 7
1688 2515 7
1689 2516 7
1690 2517 7
1691 2518 6
1692 2519 6
1693 2520 7
1694 2521 7
1695 2522 7
1696 2523 7
1697 2524 7
1698 2525 6
1699 2526 5
1700 2527 4
1701 2528 4
1702 2529 4
1703 2530 4
1704 2531 4
1705 2532 4
1706 2533 5
1707 2534 5
1708 2535 5
1709 2536 5
1710 2537 5
1711 2538 6
1712 2539 6
1713 2540 6
1714 2541 6
1715 2542 7
1716 2543 7
1717 2544 7
1718 2545 7
1719 2546 7
1720 2547 7
1721 2548 6
1722 2549 5
1723 2550 5
1724 2551 6
1725 2552 6
1726 2553 6
1727 2554 6
1728 2555 6
1729 2556 6
1730 2557 6
1731 2558 5
1732 2559 5
1733 2560 6

P
P
P

```
BEGIN
queue_desc [0] =
  (.id + .id [acr$w_queue])<0,8>;
queue_desc [1] =
  id + .id [acr$w_queue] + 1;
END;
if (.version EQL acr$k_version3) then
BEGIN
  if (.id [acr$w_nodeaddr] NEQ 0) then
  BEGIN
    address_desc [0] =
      (.id + .id [acr$w_nodeaddr])<0,8>;
    address_desc [1] =
      id + .id [acr$w_nodeaddr] + 1;
  END;
  if (.id [acr$w_remoteid] NEQ 0) then
  BEGIN
    remote_id_desc [0] =
      (.id + .id [acr$w_remoteid])<0,8>;
    remote_id_desc [1] =
      id + .id [acr$w_remoteid] + 1;
  END;
  END;
END;
[acr$k_resource]:
  Resource packets
  BEGIN
  BIND
    resource = .packet:      bblock;

  BEGIN
    start_time_desc [0] = 8;
    start_time_desc [1] = resource [acr$q_login];
    if .finish_time_desc [0] NEQ 0 then
    BEGIN
      perform (lib$subx (
        .finish_time_desc [1],
        .start_time_desc [1],
        elapsed time, %ref(2)));
      elapsed_desc [0] = 8;
    END;
  END;

  BEGIN
    status_desc [0] = 4;
    status_desc [1] = resource [acr$l_status];
    status_text_desc [0] = 80;
    status_text_desc [1] = status_text_buffer;
    translate_status
      (.status_desc [1], status_text_desc);
  END;
  BEGIN
```

```
: 1734      2561  6      execution_desc [0] = 4;
: 1735      2562  6      execution_desc [1] = resource [acr$l_imgcnt];
: 1736      2563  5      END;
: 1737      2564  5
: 1738      2565  6      BEGIN
: 1739      2566  6      BUILTIN emul;
: 1740      2567  6
: 1741      2568  6      cpu_time_desc [0] = 4;
: 1742      2569  6      cpu_time_desc [1] = resource [acr$l_cputime];
: 1743      2570  6      emul (%ref(100000), resource [acr$l_cputime],
: 1744      2571  6          %ref(0),quad_cpu_time [0]);
: 1745      2572  6      END;
: 1746      2573  5
: 1747      2574  5      BEGIN
: 1748      2575  6      page_faults_desc [0] = 4;
: 1749      2576  6      page_faults_desc [1] = resource [acr$l_faults];
: 1750      2577  6      END;
: 1751      2578  5
: 1752      2579  5      BEGIN
: 1753      2580  6      page_reads_desc [0] = 4;
: 1754      2581  6      page_reads_desc [1] = resource [acr$l_faultio];
: 1755      2582  6      END;
: 1756      2583  5
: 1757      2584  5      BEGIN
: 1758      2585  6      working_set_desc [0] = 4;
: 1759      2586  6      working_set_desc [1] = resource [acr$l_wspeak];
: 1760      2587  6      END;
: 1761      2588  5
: 1762      2589  5      BEGIN
: 1763      2590  6      page_file_desc [0] = 4;
: 1764      2591  6      page_file_desc [1] = resource [acr$l_pagefl];
: 1765      2592  6      END;
: 1766      2593  5
: 1767      2594  5      BEGIN
: 1768      2595  6      direct_io_desc [0] = 4;
: 1769      2596  6      direct_io_desc [1] = resource [acr$l_diocnt];
: 1770      2597  6      END;
: 1771      2598  5
: 1772      2599  5      BEGIN
: 1773      2600  6      buffered_io_desc [0] = 4;
: 1774      2601  6      buffered_io_desc [1] = resource [acr$l_biocnt];
: 1775      2602  6      END;
: 1776      2603  5
: 1777      2604  5      BEGIN
: 1778      2605  6      volumes_desc [0] = 4;
: 1779      2606  6      volumes_desc [1] = resource [acr$l_volumes];
: 1780      2607  6      END;
: 1781      2608  5      END;
: 1782      2609  4      [acr$k_imagename]:
: 1783      2610  4      |
: 1784      2611  4      | Imagename packets
: 1785      2612  4      |
: 1786      2613  4      |
: 1787      2614  4      |
: 1788      2615  5      BEGIN
: 1789      2616  5
: 1790      2617  5      BIND
```


1791 2618 5
1792 2619 5
1793 2620 5
1794 2621 5
1795 2622 5
1796 2623 6
1797 2624 6
1798 2625 6
1799 2626 6
1800 2627 6
1801 2628 6
1802 2629 6
1803 2630 6
1804 2631 6
1805 2632 6
1806 2633 6
1807 2634 6
1808 2635 6
1809 2636 6
1810 2637 6
1811 2638 6
1812 2639 6
1813 2640 6
1814 2641 6
1815 2642 6
1816 2643 6
1817 2644 6
1818 2645 6
1819 2646 6
1820 2647 6
1821 2648 6
1822 2649 6
1823 2650 6
1824 2651 6
1825 2652 6
1826 2653 6
1827 2654 6
1828 2655 6
1829 2656 6
1830 2657 6
1831 2658 6
1832 2659 6
1833 2660 6
1834 2661 6
1835 2662 5
1836 2663 4
1837 2664 4
1838 2665 4
1839 2666 4
1840 2667 4
1841 2668 4
1842 2669 5
1843 2670 5
1844 2671 5
1845 2672 5
1846 2673 5
1847 2674 5

```
image = .packet:      bblock;
LOCAL
  name_end;
if (.(image [acr$imagename])<0,8> NEQ 0) then
  BEGIN
    MACRO
      ITMLST$K_LENGTH = 8 %,      ! Item list member size.
      ITMLST$W_RETLN = 0,0,14 0 %, ! Return length.
      ITMLST$W_ITMCD = 2,0,16,0 %, ! Item code.
      ITMLST$A_BUFADR = 4,0,32,0 % ! Address of component.
    ;
    LOCAL
      itmlst: block[ ( itmlst$k_length * 1 ) +
        %upval, byte],
        ! $FILESCAN item list (plus ending zero)
      status;
    MAP
      itmlst: blockvector[, itmlst$k_length, byte];
    image_file_desc [0] =
      (image [acr$imagename])<0,8>;
    image_file_desc [1] =
      image [acr$imagename] + 1;
    ;
    ! Do $FILESCAN to get filename.
    itmlst[0, itmlst$w_itmcd] = fscn$name;
    itmlst[1, 0, 0, 32, 0] = 0;
    status = $filescan(
      srcstr = image_file_desc,
      valuelst = itmlst );
    if not .status
    then return .status;
    subtype_desc [0] = image_desc[0] =
      itmlst[0, itmlst$w_retlst]; ! Length
    subtype_desc [1] = image_desc[1] =
      itmlst[0, itmlst$a_bufadr]; ! Address
  END;
END;
[acr$k_filename]:
  ! Filename packets
  BEGIN
  BIND
    file = .packet: bblock;
  if (.(file [acr$filename])<0,8> NEQ 0) then
```

```

: 1848      2675 6      BEGIN
: 1849      2676 6      file_desc [0] = .(file [acr$t_filename])<0,8>;
: 1850      2677 6      file_desc [1] = file [acr$t_filename] + 1;
: 1851      2678 5      END;
: 1852      2679 5      END;
: 1853      2680 4      [acr$k_user_data]:
: 1854      2681 4      . User data packets
: 1855      2682 4      BEGIN
: 1856      2683 4      BIND
: 1857      2684 4      user_data = .packet: bblock;
: 1858      2685 5      if (.(user_data [acr$t_user_data])<0,8> NEQ 0) then
: 1859      2686 5      BEGIN
: 1860      2687 5      user_data_desc [0] =
: 1861      2688 5      .(user_data [acr$t_user_data])<0,8>;
: 1862      2689 5      user_data_desc [1] =
: 1863      2690 5      user_data [acr$t_user_data] + 1;
: 1864      2691 6      END;
: 1865      2692 6      END;
: 1866      2693 6      [acr$k_print]:
: 1867      2694 6      . Print resource packets
: 1868      2695 6      BEGIN
: 1869      2696 5      BIND
: 1870      2697 4      print = .packet: bblock;
: 1871      2698 4      BEGIN
: 1872      2699 4      status_desc [0] = 4;
: 1873      2700 4      status_desc [1] = print [acr$l_printsts];
: 1874      2701 4      status_text_desc [0] = 80;
: 1875      2702 4      status_text_desc [1] = status_text_buffer;
: 1876      2703 5      translate_status
: 1877      2704 5      (.status_desc [1], status_text_desc);
: 1878      2705 5      END;
: 1879      2706 5      if (.(print [acr$q_begtime])<0,32> NEQ 0) OR
: 1880      2707 5      (.(print [acr$q_begtime] + 4)<0,32> NEQ 0) then
: 1881      2708 6      BEGIN
: 1882      2709 6      start_time_desc [0] = 8;
: 1883      2710 6      start_time_desc [1] = print [acr$q_begtime];
: 1884      2711 6      if .finish_time_desc [0] NEQ 0 then
: 1885      2712 6      BEGIN
: 1886      2713 6      perform (lib$subx (
: 1887      2714 6      .finish_time_desc [1],
: 1888      2715 5      .start_time_desc [1],
: 1889      2716 5      elapsed_time, %ref(2)));
: 1890      2717 5      elapsed_desc [0] = 8;
: 1891      2718 6      END;
: 1892      2719 6      else if (.(print [acr$q_quetime])<0,32> NEQ 0) OR
: 1893      2720 6      (.(print [acr$q_quetime] + 4)<0,32> NEQ 0) then
: 1894      2721 6      BEGIN
: 1895      2722 7      perform (lib$subx (
: 1896      2723 7      .finish_time_desc [1],
: 1897      2724 7      .start_time_desc [1],
: 1898      2725 7      elapsed_time, %ref(2)));
: 1899      2726 7      elapsed_desc [0] = 8;
: 1900      2727 7      END;
: 1901      2728 6      END
: 1902      2729 6      else if (.(print [acr$q_quetime])<0,32> NEQ 0) OR
: 1903      2730 5      (.(print [acr$q_quetime] + 4)<0,32> NEQ 0) then
: 1904      2731 5
```

```

: 1905      2732 6      BEGIN
: 1906      2733 6      queue_time_desc [0] = 8;
: 1907      2734 6      queue_time_desc [1] = print [acr$q_quetime];
: 1908      2735 6      if .finish_time_desc [0] NEQ 0 then
: 1909      P 2736 6      perform (lib$subx (
: 1910      P 2737 6      .finish_time_desc [1],
: 1911      P 2738 6      .queue_time_desc [1],
: 1912      2739 6      elapsed_time, %ref(2));
: 1913      2740 5      END;
: 1914      2741 5
: 1915      2742 6      BEGIN
: 1916      2743 6      BUILTIN emul;
: 1917      2744 6
: 1918      2745 6      cpu_time_desc [0] = 4;
: 1919      2746 6      cpu_time_desc [1] = print [acr$l_symcputim];
: 1920      2747 6      emul (%ref(100000), print [acr$l_symcputim],
: 1921      2748 6      %ref(0), quad_cpu_time [0]);
: 1922      2749 6      END;
: 1923      2750 5
: 1924      2751 5      BEGIN
: 1925      2752 6      sym_page_desc [0] = 4;
: 1926      2753 6      sym_page_desc [1] = print [acr$l_pagecnt];
: 1927      2754 6      END;
: 1928      2755 5
: 1929      2756 5      BEGIN
: 1930      2757 6      sym_qio_desc [0] = 4;
: 1931      2758 6      sym_qio_desc [1] = print [acr$l_qiocnt];
: 1932      2759 6      END;
: 1933      2760 5
: 1934      2761 5      BEGIN
: 1935      2762 6      sym_get_desc [0] = 4;
: 1936      2763 6      sym_get_desc [1] = print [acr$l_getcnt];
: 1937      2764 6      END;
: 1938      2765 5      END;
: 1939      2766 4
: 1940      2767 4      [OTHERWISE]:
: 1941      2768 4      : Unknown packet type
: 1942      2769 4      :
: 1943      2770 4      : return false;
: 1944      2771 4
: 1945      2772 4
: 1946      2773 4
: 1947      2774 4      TES;
: 1948      2775 3      END;
: 1949      2776 3      packet = .packet + .packet [acr$w_length];
: 1950      2777 2      END;
: 1951      2778 2
: 1952      2779 2
: 1953      2780 2      return true;
: 1954      2781 2
: 1955      2782 1      END;
```

```

00 45 56 49 54 43 41 52 45 54 4E 49 00983 .BLKB 1
00 00 53 53 45 43 4F 52 50 42 55 53 00984 P.AAW: .ASCII \INTERACTIVE\<0>
00 00 53 53 45 43 4F 52 50 42 55 53 009C0 P.AAX: .ASCII \SUBPROCESS\<0><0>
```

61	6E	69	6D	72	65	54	44	45	48	43	41	54	45	44	009CC	P.AAY:	.ASCII	\DETACHED\
							00	00	00	48	43	54	41	42	009D4	P.AAZ:	.ASCII	\BATCH\<0><0><0>
							00	48	52	4F	57	54	45	4E	009DC	P.ABA:	.ASCII	\NETWORK\<0>
							00	53	53	45	43	4F	52	50	009E4	P.ABB:	.ASCII	\PROCESS\<0>
							20	73	73	65	63	6F	72	50	009EC	P.ABC:	.ASCII	\Process Termination\<0>
							00	00	00	45	47	41	4D	49	009FB			
69	74	61	6E	69	6D	72	65	54	20	65	67	61	6D	49	00A00	P.ABD:	.ASCII	\IMAGE\<0><0><0>
							00	00	00	00	00	00	6E	6F	00A08	P.ABE:	.ASCII	\Image Termination\<0><0><0>
							00	54	49	4E	49	53	59	53	00A17			
49	4C	41	49	54	49	4E	49	20	4D	45	54	53	59	53	00A1C	P.ABF:	.ASCII	\SYSINIT\<0>
						00	00	00	4E	4F	49	54	41	5A	00A24	P.ABG:	.ASCII	\SYSTEM INITIALIZATION\<0><0><0>
							00	45	4D	49	54	54	45	53	00A33			
61	68	43	20	65	6D	69	54	20	6D	65	74	73	79	53	00A3C	P.ABH:	.ASCII	\SETTIME\<0>
							00	00	00	00	00	65	67	6E	00A44	P.ABI:	.ASCII	\System Time Change\<0><0>
							00	4C	49	41	46	47	4F	4C	00A53			
00	00	45	52	55	4C	49	41	46	20	4E	49	47	4F	4C	00A58	P.ABJ:	.ASCII	\LOGFAIL\<0>
							00	00	00	54	4E	49	52	50	00A60	P.ABK:	.ASCII	\LOGIN FAILURE\<0><0><0>
							00	00	00	54	4E	49	52	50	00A6F			
69	6D	72	65	54	20	62	6F	4A	20	54	4E	49	52	50	00A70	P.ABL:	.ASCII	\PRINT\<0><0><0>
						00	00	00	6E	6F	69	74	61	6E	00A78	P.ABM:	.ASCII	\PRINT Job Termination\<0><0><0>
							00	00	00	54	4E	49	52	50	00A87			
							00	00	00	54	4E	49	52	50	00A90	P.ABN:	.ASCII	\USER\
							00	00	00	54	4E	49	52	50	00A94	P.ABO:	.ASCII	\USER Message\
69	74	6E	75	6F	63	63	41	20	65	6C	62	61	6E	45	00AA0	P.ABP:	.ASCII	\ENABLE\<0><0>
							00	00	00	00	00	00	67	6E	00AA8	P.ABQ:	.ASCII	\Enable Accounting\<0><0><0>
							00	45	4C	42	41	53	49	44	00AB7			
74	6E	75	6F	63	63	41	20	65	6C	62	61	73	69	44	00ABC	P.ABR:	.ASCII	\DISABLE\<0>
							00	00	00	00	00	67	6E	69	00AC4	P.ABS:	.ASCII	\Disable Accounting\<0><0>
							00	00	00	43	41	54	4C	41	00AD3			
75	6F	63	63	41	20	65	74	61	6E	72	65	74	6C	41	00AD8	P.ABT:	.ASCII	\ALTACM\<0><0>
							00	4C	42	5F	45	4C	49	46	00AE0	P.ABU:	.ASCII	\Alternate Accounting Manager\
				</														

01FC 00000 PROCESS_V3_RECORD:

TYPE	PROCESS	VS	RECORD	ADDRESS	DATA	OPERATION	COMMENT	PC
58	00000000G	00	9E	00002	MOVAB	Save R2,R3,R4,R5,R6,R7,R8	2189	
57	0000'	CF	9E	00009	MOVAB	TRANSLATE STATUS, R8		
56	00000000G	00	9E	0000E	MOVAB	PACKET, R7		
55	FE0B'	CF	9E	00015	MOVAB	LIBSSUBX, R6		
54	00000000'	EF	9E	0001A	MOVAB	P.AAW, R5		
5E		10	C2	00021	SUBL2	TYPE_DESC, R4		
53	04	AC	D0	00024	MOVL	#16, -SP	2231	
03		63	E9	00028	BLBC	BUFFER, R3		
		0564	31	0002B	BRW	(R3), 1\$		
07		01	EF	0002E	EXTZV	57\$	2233	
04		00	EF	00035	EXTZV	#1, #7, (R3), RECORD_TYPE	2234	
01		0F	EF	0003D	EXTZV	#0, #4, 1(R3), RECORD SUBTYPE	2235	
03	FECO	C4	E9	00044	BLBC	#15, #1, (R3), CUSTOMER	2236	
		01BD	31	00049	BRW	CUSTOMER, 2\$		
51	FECB	C4	D0	0004C	MOVL	20\$	2239	
52		01	D0	00051	MOVL	RECORD_TYPE, R1		
		51	D5	00054	TSTL	#1, R2	2241	
		59	13	00056	BEQL	R1		
03		51	D1	00058	CMPL	7\$		
		54	1A	0005B	BGTRU	R1, #3		
		52	D4	0005D	CLRL	7\$		
50	FECB	C4	D0	0005F	MOVL	R2	2243	
01		50	D1	00064	CMPL	RECORD_SUBTYPE, R0	2245	
		0A	12	00067	BNEQ	R0, #1		
A0	A4	0B	D0	00069	MOVL	3\$	2247	
A4	A4	65	9E	0006D	MOVAB	#11, PROCESS_DESC	2248	
		3E	11	00071	BRB	P.AAW, PROCESS_DESC+4	2243	
02		50	D1	00073	CMPL	7\$	2250	
		0B	12	00076	BNEQ	R0, #2		
A0	A4	0A	D0	00078	MOVL	4\$	2252	
A4	A4	0C	A5	0007C	MOVAB	#10, PROCESS_DESC	2253	
		2E	11	00081	BRB	P.AAX, PROCESS_DESC+4	2243	
03		50	D1	00083	CMPL	7\$	2255	
		0B	12	00086	BNEQ	R0, #3		
A0	A4	08	D0	00088	MOVL	5\$	2257	
A4	A4	18	A5	0008C	MOVAB	#8, PROCESS_DESC	2258	
		1E	11	00091	BRB	P.AAY, PROCESS_DESC+4	2243	
04		50	D1	00093	CMPL	7\$	2260	
		0B	12	00096	BNEQ	R0, #4		
A0	A4	05	D0	00098	MOVL	6\$	2262	
A4	A4	20	A5	0009C	MOVAB	#5, PROCESS_DESC	2263	
		0E	11	000A1	BRB	P.AAZ, PROCESS_DESC+4	2243	
05		50	D1	000A3	CMPL	7\$	2265	
		09	12	000A6	BNEQ	R0, #5		
A0	A4	07	D0	000AB	MOVL	7\$	2267	
A4	A4	28	A5	000AC	MOVAB	#7, PROCESS_DESC	2268	
		51	D5	000B1	TSTL	P.ABA, PROCESS_DESC+4	2272	
		1D	13	000B3	BEQL	R1		
02		51	D1	000B5	CMPL	8\$		
		18	1A	000B8	BGTRU	R1, #2		
		52	D4	000BA	CLRL	8\$		
		07	D0	000BC	MOVL	R2	2274	
04	A4	30	A5	000BF	MOVAB	#7, TYPE_DESC	2275	
08	A4	13	D0	000C4	MOVL	P.ABB, TYPE_DESC+4	2276	
					MOVL	#19, TYPE_TEXT_DESC		

0C	A4	38	A5	9E	000C8	MOVAB	P.ABC, TYPE_TEXT_DESC+4	2277	
D8	A4	A0	A4	7D	000CD	MOVQ	PROCESS_DESC, SUBTYPE_DESC	2278	
	03		51	D1	000D2	8\$:	CMPL	R1, #3	2281
			18	1F	000D5		BLSSU	9\$	
	04		51	D1	000D7		CMPL	R1, #4	
			13	1A	000DA		BGTRU	9\$	
			52	D4	000DC		CLRL	R2	
	64		05	D0	000DE		MOVL	#5, TYPE_DESC	2283
04	A4	4C	A5	9E	000E1	MOVAB	P.ABD, TYPE_DESC+4	2284	
08	A4		11	D0	000E6	MOVL	#17, TYPE_TEXT_DESC	2285	
0C	A4	54	A5	9E	000EA	MOVAB	P.ABE, TYPE_TEXT_DESC+4	2286	
	05		51	D1	000EF	9\$:	CMPL	R1, #5	2288
			13	12	000F2		BNEQ	10\$	
			52	D4	000F4		CLRL	R2	
	64		07	D0	000F6		MOVL	#7, TYPE_DESC	2290
04	A4	68	A5	9E	000F9	MOVAB	P.ABF, TYPE_DESC+4	2291	
08	A4		15	D0	000FE	MOVL	#21, TYPE_TEXT_DESC	2292	
0C	A4	70	A5	9E	00102	MOVAB	P.ABG, TYPE_TEXT_DESC+4	2293	
	06		51	D1	00107	10\$:	CMPL	R1, #6	2295
			15	12	0010A		BNEQ	11\$	
			52	D4	0010C		CLRL	R2	
	64		07	D0	0010E		MOVL	#7, TYPE_DESC	2297
04	A4	0088	C5	9E	00111	MOVAB	P.ABH, TYPE_DESC+4	2298	
08	A4		12	D0	00117	MOVL	#18, TYPE_TEXT_DESC	2299	
0C	A4	0090	C5	9E	0011B	MOVAB	P.ABI, TYPE_TEXT_DESC+4	2300	
	07		51	D1	00121	11\$:	CMPL	R1, #7	2302
			15	12	00124		BNEQ	12\$	
			52	D4	00126		CLRL	R2	
	64		07	D0	00128		MOVL	#7, TYPE_DESC	2304
04	A4	00A4	C5	9E	0012B	MOVAB	P.ABJ, TYPE_DESC+4	2305	
08	A4		0D	D0	00131	MOVL	#13, TYPE_TEXT_DESC	2306	
0C	A4	00AC	C5	9E	00135	MOVAB	P.ABK, TYPE_TEXT_DESC+4	2307	
	08		51	D1	0013B	12\$:	CMPL	R1, #8	2309
			15	12	0013E		BNEQ	13\$	
			52	D4	00140		CLRL	R2	
	64		05	D0	00142		MOVL	#5, TYPE_DESC	2311
04	A4	00BC	C5	9E	00145	MOVAB	P.ABL, TYPE_DESC+4	2312	
08	A4		15	D0	0014B	MOVL	#21, TYPE_TEXT_DESC	2313	
0C	A4	00C4	C5	9E	0014F	MOVAB	P.ABM, TYPE_TEXT_DESC+4	2314	
	09		51	D1	00155	13\$:	CMPL	R1, #9	2316
			15	12	00158		BNEQ	14\$	
			52	D4	0015A		CLRL	R2	
	64		04	D0	0015C		MOVL	#4, TYPE_DESC	2318
04	A4	00DC	C5	9E	0015F	MOVAB	P.ABN, TYPE_DESC+4	2319	
08	A4		0C	D0	00165	MOVL	#12, TYPE_TEXT_DESC	2320	
0C	A4	00E0	C5	9E	00169	MOVAB	P.ABO, TYPE_TEXT_DESC+4	2321	
	0A		51	D1	0016F	14\$:	CMPL	R1, #10	2323
			15	12	00172		BNEQ	15\$	
			52	D4	00174		CLRL	R2	
	64		06	D0	00176		MOVL	#6, TYPE_DESC	2325
04	A4	00EC	C5	9E	00179	MOVAB	P.ABP, TYPE_DESC+4	2326	
08	A4		11	D0	0017F	MOVL	#17, TYPE_TEXT_DESC	2327	
0C	A4	00F4	C5	9E	00183	MOVAB	P.ABQ, TYPE_TEXT_DESC+4	2328	
	0B		51	D1	00189	15\$:	CMPL	R1, #11	2330
			15	12	0018C		BNEQ	16\$	
			52	D4	0018E		CLRL	R2	
	64		07	D0	00190		MOVL	#7, TYPE_DESC	2332

04	A4	0108	C5	9E	00193	MOVAB	P.ABR, TYPE_DESC+4	2333
08	A4		12	D0	00199	MOVL	#18, TYPE_TEXT_DESC	2334
0C	A4	0110	C5	9E	0019D	MOVAB	P.ABS, TYPE_TEXT_DESC+4	2335
	0C		51	D1	001A3	16\$: CMPL	R1, #12	2337
			15	12	001A6	BNEQ	17\$	
			52	D4	001A8	CLRL	R2	
	64		06	D0	001AA	MOVL	#6, TYPE_DESC	2339
04	A4	0124	C5	9E	001AD	MOVAB	P.ABT, TYPE_DESC+4	2340
08	A4		1C	D0	001B3	MOVL	#28, TYPE_TEXT_DESC	2341
0C	A4	012C	C5	9E	001B7	MOVAB	P.ABU, TYPE_TEXT_DESC+4	2342
	0E		51	D1	001BD	17\$: CMPL	R1, #14	2344
			15	12	001C0	BNEQ	18\$	
			52	D4	001C2	CLRL	R2	
	64		04	D0	001C4	MOVL	#4, TYPE_DESC	2346
04	A4	0148	C5	9E	001C7	MOVAB	P.ABV, TYPE_DESC+4	2347
08	A4		20	D0	001CD	MOVL	#32, TYPE_TEXT_DESC	2348
0C	A4	0150	C5	9E	001D1	MOVAB	P.ABW, TYPE_TEXT_DESC+4	2349
	0D		51	D1	001D7	18\$: CMPL	R1, #13	2352
			15	12	001DA	BNEQ	19\$	
			52	D4	001DC	CLRL	R2	
	64		04	D0	001DE	MOVL	#4, TYPE_DESC	2354
04	A4	0170	C5	9E	001E1	MOVAB	P.ABX, TYPE_DESC+4	2355
08	A4		1F	D0	001E7	MOVL	#31, TYPE_TEXT_DESC	2356
0C	A4	0178	C5	9E	001EB	MOVAB	P.ABY, TYPE_TEXT_DESC+4	2357
	28		52	E9	001F1	19\$: BLBC	R2, 21\$	2360
	64		07	D0	001F4	MOVL	#7, TYPE_DESC	2362
04	A4	0198	C5	9E	001F7	MOVAB	P.ABZ, TYPE_DESC+4	2363
08	A4		13	D0	001FD	MOVL	#19, TYPE_TEXT_DESC	2364
0C	A4	01A0	C5	9E	00201	MOVAB	P.ACA, TYPE_TEXT_DESC+4	2365
			13	11	00207	BRB	21\$	2366
	64		08	D0	00209	20\$: MOVL	#8, TYPE_DESC	2371
04	A4	01B4	C5	9E	0020C	MOVAB	P.ACB, TYPE_DESC+4	2372
08	A4		14	D0	00212	MOVL	#20, TYPE_TEXT_DESC	2373
0C	A4	01BC	C5	9E	00216	MOVAB	P.ACC, TYPE_TEXT_DESC+4	2374
FEC4	C4	02	A3	3C	0021C	21\$: MOVZWL	2(R3), RECORD_LENGTH	2379
		04	A3	D5	00222	TSTL	4(R3)	2383
			05	12	00225	BNEQ	22\$	
		08	A3	D5	00227	TSTL	8(R3)	2384
			08	13	0022A	BEQL	23\$	
FF40	C4		08	D0	0022C	22\$: MOVL	#8, FINISH_TIME_DESC	2387
FF44	C4	04	A3	9E	00231	MOVAB	4(R3), FINISH_TIME_DESC+4	2388
	67	0C	A3	9E	00237	23\$: MOVAB	12(R3), PACKET	2398
	50		67	D0	0023B	24\$: MOVL	PACKET, R0	2399
51	53	FEC4	C4	C1	0023E	ADDL3	RECORD_LENGTH, R3, R1	
	51		50	D1	00244	CMPL	R0, R1	
			03	1F	00247	BLSSU	25\$	
			0342	31	00249	BRW	56\$	
	03		60	E8	0024C	25\$: BLBS	(R0), 26\$	2404
			0340	31	0024F	BRW	57\$	
			60	B5	00252	26\$: TSTW	(R0)	2405
			03	18	00254	BGEQ	27\$	
			0328	31	00256	BRW	55\$	
01	60	07	01	ED	00259	27\$: CMPZV	#1, #7, (R0), #1	2408
			03	13	0025E	BEQL	28\$	
			011D	31	00260	BRW	40\$	
	52		50	D0	00263	28\$: MOVL	R0, R2	2415
FF48	C4		04	D0	00266	MOVL	#4, IDENT_DESC	2418

FF4C	C4	04	A2	9E	0026B	MOVAB	4(R2), IDENT_DESC+4	2419
		08	A2	D5	00271	TSTL	8(R2)	2422
			0B	13	00274	BEQL	29\$	
FF70	C4		04	D0	00276	MOVL	#4, OWNER_DESC	2424
FF74	C4	08	A2	9E	0027B	MOVAB	8(R2), OWNER_DESC+4	2425
10	A4		04	D0	00281	MOVL	#4, UIC_BINARY_DESC	2429
14	A4	0C	A2	9E	00285	MOVAB	12(R2), UIC_BINARY_DESC+4	2430
18	A4		02	D0	0028A	MOVL	#2, UIC_GROUP_DESC	2431
1C	A4	0E	A2	9E	0028E	MOVAB	14(R2), UIC_GROUP_DESC+4	2432
20	A4		02	D0	00293	MOVL	#2, UIC_MEMBER_DESC	2433
24	A4	0C	A2	9E	00297	MOVAB	12(R2), UIC_MEMBER_DESC+4	2434
	23	0368	C4	E9	0029C	BLBC	SUMMARY_UIC, 30\$	2441
28	A4		8F	9A	002A1	MOVZBL	#67, UIC_DESC	2444
2C	A4	0104	C4	9E	002A6	MOVAB	UIC_BUF, UIC_DESC+4	2445
			0C	A2	DD	PUSHL	12(R2)	2449
			28	A4	9F	PUSHAB	UIC_DESC	
			28	A4	9F	PUSHAB	UIC_DESC	
		01D4	C5	9F	002B5	PUSHAB	P.ACD	
00000000G	00		04	FB	002B9	CALLS	#4, LIB\$SYS_FAO	
	01		50	E8	002C0	BLBS	STATUS, 30\$	
			04	002C3	RET			
98	A4		08	D0	002C4	MOVL	#8, PRIVILEGE_DESC	2454
9C	A4	10	A2	9E	002C8	MOVAB	16(R2), PRIVILEGE_DESC+4	2455
90	A4		01	D0	002CD	MOVL	#1, PRIORITY_DESC	2459
94	A4	18	A2	9E	002D1	MOVAB	24(R2), PRIORITY_DESC+4	2460
	50	1A	A2	3C	002D6	MOVZWL	26(R2), R0	2463
			0B	13	002DA	BEQL	31\$	
38	A4		6042	9A	002DC	MOVZBL	(R0)[R2], USER_DESC	2466
3C	A4	01	A042	9E	002E1	MOVAB	1(R0)[R2], USER_DESC+4	2468
	50	1C	A2	3C	002E7	MOVZWL	28(R2), R0	2470
			0D	13	002EB	BEQL	32\$	
FEF8	C4		6042	9A	002ED	MOVZBL	(R0)[R2], ACCOUNT_DESC	2473
FEFC	C4	01	A042	9E	002F3	MOVAB	1(R0)[R2], ACCOUNT_DESC+4	2475
	50	1E	A2	3C	002FA	MOVZWL	30(R2), R0	2477
			0D	13	002FE	BEQL	33\$	
FF68	C4		6042	9A	00300	MOVZBL	(R0)[R2], NODE_DESC	2480
FF6C	C4	01	A042	9E	00306	MOVAB	1(R0)[R2], NODE_DESC+4	2482
	50	20	A2	3C	0030D	MOVZWL	32(R2), R0	2484
			0B	13	00311	BEQL	34\$	
F8	A4		6042	9A	00313	MOVZBL	(R0)[R2], TERMINAL_DESC	2487
FC	A4	01	A042	9E	00318	MOVAB	1(R0)[R2], TERMINAL_DESC+4	2489
	50	22	A2	3C	0031E	MOVZWL	34(R2), R0	2491
			0D	13	00322	BEQL	35\$	
FF60	C4		6042	9A	00324	MOVZBL	(R0)[R2], JOB_DESC	2494
FF64	C4	01	A042	9E	0032A	MOVAB	1(R0)[R2], JOB_DESC+4	2496
		24	A2	D5	00331	TSTL	36(R2)	2498
			0B	13	00334	BEQL	36\$	
FF20	C4		04	D0	00336	MOVL	#4, ENTRY_DESC	2500
FF24	C4	24	A2	9E	0033B	MOVAB	36(R2), ENTRY_DESC+4	2501
	50	28	A2	3C	00341	MOVZWL	40(R2), R0	2503
			0B	13	00345	BEQL	37\$	
A8	A4		6042	9A	00347	MOVZBL	(R0)[R2], QUEUE_DESC	2506
AC	A4	01	A042	9E	0034C	MOVAB	1(R0)[R2], QUEUE_DESC+4	2508
	02	FED0	C4	D1	00352	CMPL	VERSION, #2	2510
			24	12	00357	BNEQ	39\$	
	50	2A	A2	3C	00359	MOVZWL	42(R2), R0	2512
			0D	13	0035D	BEQL	38\$	

				FF00	C4		6042	9A	0035F	MOVZBL	(R0)[R2], ADDRESS_DESC	2515	
				FF04	C4	01	A042	9E	00365	MOVAB	1(R0)[R2], ADDRESS_DESC+4	2517	
					50	2C	A2	3C	0036C	MOVZWL	44(R2), R0	2519	
							0B	13	00370	BEQL	39\$		
				B8	A4		6042	9A	00372	MOVZBL	(R0)[R2], REMOTE_ID_DESC	2522	
				BC	A4	01	A042	9E	00377	MOVAB	1(R0)[R2], REMOTE_ID_DESC+4	2524	
02		60					0201	31	0037D	BRW	55\$	2406	
					07		01	ED	00380	CMPZV	#1, #7, (R0), #2	2529	
							03	13	00385	BEQL	41\$		
							00B4	31	00387	BRW	44\$		
					52		50	D0	0038A	MOVL	R0, R2	2536	
				C0	A4		08	D0	0038D	MOVL	#8, START_TIME_DESC	2539	
				C4	A4	04	A2	9E	00391	MOVAB	4(R2), START_TIME_DESC+4	2540	
						FF40	C4	D5	00396	TSTL	FINISH_TIME_DESC	2541	
							1C	13	0039A	BEQL	43\$		
					6E		02	D0	0039C	MOVL	#2, (SP)	2546	
							5E	DD	0039F	PUSHL	SP		
						FEE8	C4	9F	003A1	PUSHAB	ELAPSED_TIME		
						C4	A4	DD	003A5	PUSHL	START_TIME_DESC+4		
						FF44	C4	DD	003AB	PUSHL	FINISH_TIME_DESC+4		
					66		04	FB	003AC	CALLS	#4, LIB\$SUBX		
					01		50	E8	003AF	BLBS	STATUS, 42\$		
							04	003B2	RET				
				FF28	C4		08	D0	003B3	MOVL	#8, ELAPSED_DESC	2547	
				C8	A4		04	D0	003B8	MOVL	#4, STATUS_DESC	2552	
				CC	A4	0C	A2	9E	003BC	MOVAB	12(R2), STATUS_DESC+4	2553	
				D0	A4	50	8F	9A	003C1	MOVZBL	#80, STATUS_TEXT_DESC	2554	
				D4	A4	00B0	C4	9E	003C6	MOVAB	STATUS_TEXT_BUFFER, STATUS_TEXT_DESC+4	2555	
						D0	A4	9F	003CC	PUSHAB	STATUS_TEXT_DESC	2557	
						CC	B4	DD	003CF	PUSHL	@STATUS_DESC+4		
					68		02	FB	003D2	CALLS	#2, TRANSLATE_STATUS		
				FF30	C4		04	D0	003D5	MOVL	#4, EXECUTION_DESC	2561	
				FF34	C4	10	A2	9E	003DA	MOVAB	16(R2), EXECUTION_DESC+4	2562	
				FF10	C4		04	D0	003E0	MOVL	#4, CPU_TIME_DESC	2569	
				FF14	C4	14	A2	9E	003E5	MOVAB	20(R2), CPU_TIME_DESC+4	2570	
FEE0	C4	00		14	A2	000186A0	8F	7A	003EB	EMUL	#100000, 20(R2), #0, QUAD_CPU_TIME	2572	
				FF78	C4		04	D0	003F7	MOVL	#4, PAGE_FAULTS_DESC	2576	
				FF7C	C4	18	A2	9E	003FC	MOVAB	24(R2), PAGE_FAULTS_DESC+4	2577	
				88	A4		04	D0	00402	MOVL	#4, PAGE_READS_DESC	2581	
				8C	A4	1C	A2	9E	00406	MOVAB	28(R2), PAGE_READS_DESC+4	2582	
				48	A4		04	D0	0040B	MOVL	#4, WORKING_SET_DESC	2586	
				4C	A4	20	A2	9E	0040F	MOVAB	32(R2), WORKING_SET_DESC+4	2587	
				80	A4		04	D0	00414	MOVL	#4, PAGE_FILE_DESC	2591	
				84	A4	24	A2	9E	00418	MOVAB	36(R2), PAGE_FILE_DESC+4	2592	
				FF18	C4		04	D0	0041D	MOVL	#4, DIRECT_IO_DESC	2596	
				FF1C	C4	28	A2	9E	00422	MOVAB	40(R2), DIRECT_IO_DESC+4	2597	
				FF08	C4		04	D0	00428	MOVL	#4, BUFFERED_IO_DESC	2601	
				FF0C	C4	2C	A2	9E	0042D	MOVAB	44(R2), BUFFERED_IO_DESC+4	2602	
				40	A4		04	D0	00433	MOVL	#4, VOLUMES_DESC	2606	
				44	A4	30	A2	9E	00437	MOVAB	48(R2), VOLUMES_DESC+4	2607	
							7F	11	0043C	BRB	48\$	2406	
03		60			07		01	ED	0043E	CMPZV	#1, #7, (R0), #3	2611	
							48	12	00443	BNEQ	46\$		
							04	A0	95	00445	TSTB	4(R0)	2622
							73	13	00448	BEQL	48\$		
				FF50	C4	04	A0	9A	0044A	MOVZBL	4(R0), IMAGE_FILE_DESC	2642	
				FF54	C4	05	A0	9E	00450	MOVAB	5(R0), IMAGE_FILE_DESC+4	2644	

06	AE	06	B0	00456	MOVW	#6, ITMLST+2	2649
		0C	AE	D4	CLRL	ITMLST+8	2650
			7E	D4	CLRL	-(SP)	2653
		08	AE	9F	PUSHAB	ITMLST	
00000000G	00	FF50	C4	9F	PUSHAB	IMAGE FILE DESC	
	01		03	FB	CALLS	#3, SYS\$FIC\$SCAN	
			50	EB	BLBS	STATUS, 45\$	2654
			04	00470	RET		
	50	04	AE	3C	MOVZWL	ITMLST, R0	2658
FF58	C4		50	D0	MOVL	R0, IMAGE DESC	
D8	A4		50	D0	MOVL	R0, SUBTYPE DESC	2657
	50	08	AE	D0	MOVL	ITMLST+4, R0	2660
FF5C	C4		50	D0	MOVL	R0, IMAGE DESC+4	
DC	A4		50	D0	MOVL	R0, SUBTYPE DESC+4	2659
			30	11	BRB	48\$	2406
04	60	07	01	ED	CMPZV	#1, #7, (R0), #4	2665
			13	12	BNEQ	47\$	
		04	A0	95	TSTB	4(R0)	2674
			24	13	BEQL	48\$	
FF38	C4	04	A0	9A	MOVZBL	4(R0), FILE DESC	2676
FF3C	C4	05	A0	9E	MOVAB	5(R0), FILE DESC+4	2677
			16	11	BRB	48\$	2406
05	60	07	01	ED	CMPZV	#1, #7, (R0), #5	2681
			12	12	BNEQ	49\$	
		04	A0	95	TSTB	4(R0)	2690
			0A	13	BEQL	48\$	
30	A4	04	A0	9A	MOVZBL	4(R0), USER_DATA DESC	2693
34	A4	05	A0	9E	MOVAB	5(R0), USER_DATA DESC+4	2695
			00C1	31	BRW	55\$	2406
08	60	07	01	ED	CMPZV	#1, #7, (R0), #8	2699
			03	13	BEQL	50\$	
		00C8	31	004C7	BRW	57\$	
	52		50	D0	MOVL	R0, R2	2706
C8	A4		04	D0	MOVL	#4, STATUS DESC	2709
CC	A4	04	A2	9E	MOVAB	4(R2), STATUS DESC+4	2710
D0	A4	50	8F	9A	MOVZBL	#80, STATUS TEXT DESC	2711
D4	A4	00B0	C4	9E	MOVAB	STATUS TEXT BUFFER, STATUS TEXT DESC+4	2712
		D0	A4	9F	PUSHAB	STATUS TEXT DESC	2714
		CC	B4	DD	PUSHL	2 STATUS DESC+4	
	68		02	FB	CALLS	#2, TRANSLATE STATUS	
		10	A2	D5	TSTL	16(R2)	2716
			05	12	BNEQ	51\$	
		14	A2	D5	TSTL	20(R2)	2717
			2C	13	BEQL	52\$	
C0	A4		08	D0	MOVL	#8, START TIME DESC	2719
C4	A4	10	A2	9E	MOVAB	16(R2), START TIME DESC+4	2720
		FF40	C4	D5	TSTL	FINISH TIME DESC	2721
			4C	13	BEQL	54\$	
	6E		02	D0	MOVL	#2, (SP)	2726
			5E	DD	PUSHL	SP	
FEE8	C4		C4	9F	PUSHAB	ELAPSED TIME	
			C4	DD	PUSHL	START TIME DESC+4	
FF44	C4		C4	DD	PUSHL	FINISH TIME DESC+4	
	66		04	FB	CALLS	#4, LIB\$SUBX	
FF28	7B		50	E9	BLBC	STATUS, 58\$	2727
	C4		08	D0	MOVL	#8, ELAPSED DESC	2716
			2F	11	BRB	54\$	

			08	A2	D5	00520	52\$:	TSTL	8(R2)	2730
			05	12	00523			BNEQ	53\$	
			0C	A2	D5	00525		TSTL	12(R2)	2731
			25	13	00528			BEQL	54\$	
	B0	A4	08	D0	0052A	53\$:		MOVL	#8, QUEUE_TIME_DESC	2733
	B4	A4	08	A2	9E	0052E		MOVAB	8(R2), QUEUE_TIME_DESC+4	2734
			FF40	C4	D5	00533		TSTL	FINISH_TIME_DESC	2735
			16	13	00537			BEQL	54\$	
		6E	02	D0	00539			MOVL	#2, (SP)	2739
			5E	DD	0053C			PUSHL	SP	
			FEE8	C4	9F	0053E		PUSHAB	ELAPSED_TIME	
			B4	A4	DD	00542		PUSHL	QUEUE_TIME_DESC+4	
			FF44	C4	DD	00545		PUSHL	FINISH_TIME_DESC+4	
		66	04	FB	00549			CALLS	#4, LIB\$SUBR	
		45	50	E9	0054C			BLBC	STATUS, 58\$	
	FF10	C4	04	D0	0054F	54\$:		MOVL	#4, CPU_TIME_DESC	2746
	FF14	C4	18	A2	9E	00554		MOVAB	24(R2), CPU_TIME_DESC+4	2747
FEE0	C4		000186A0	8F	7A	0055A		EMUL	#100000, 24(R2), #0, QUAD_CPU_TIME	2749
	F0	A4	04	D0	00566			MOVL	#4, SYM_PAGE_DESC	2753
	F4	A4	1C	A2	9E	0056A		MOVAB	28(R2), SYM_PAGE_DESC+4	2754
	E8	A4	04	D0	0056F			MOVL	#4, SYM_QIO_DESC	2758
	EC	A4	20	A2	9E	00573		MOVAB	32(R2), SYM_QIO_DESC+4	2759
	E0	A4	04	D0	00578			MOVL	#4, SYM_GET_DESC	2763
	E4	A4	24	A2	9E	0057C		MOVAB	36(R2), SYM_GET_DESC+4	2764
		50	67	D0	00581	55\$:		MOVL	PACKET, R0	2776
		51	02	A0	3C	00584		MOVZWL	2(R0), R1	
		67	51	C0	00588			ADDL2	R1, PACKET	
			FCAD	31	0058B			BRW	24\$	2399
		50	01	D0	0058E	56\$:		MOVL	#1, R0	2780
				04	00591			RET		
			50	D4	00592	57\$:		CLRL	R0	2782
				04	00594	58\$:		RET		

; Routine Size: 1429 bytes, Routine Base: CODE + 0B90

```
1957 2783 1 ROUTINE RECORD_SELECTED (BUFFER) =
1958 2784 1
1959 2785 1 ----
1960 2786 1
1961 2787 1 Functional description
1962 2788 1
1963 2789 1 This routine tests the record pointed to by BUFFER against
1964 2790 1 user specified selection criteria. If any test is failed
1965 2791 1 then an immediate failed return is made. Else true is returned.
1966 2792 1
1967 2793 1 Inputs:
1968 2794 1
1969 2795 1 BUFFER = Address of the current record
1970 2796 1
1971 2797 1 Outputs:
1972 2798 1
1973 2799 1 True or false. There are no error exits/returns from this routine.
1974 2800 1
1975 2801 1 ---
1976 2802 1
1977 2803 2 BEGIN
1978 2804 2
1979 2805 2 MAP buffer: ref bblock; ! Define buffer block format
1980 2806 2
1981 2807 2 MACRO
1982 2808 2 SELECT_OR_REJECT (name) =
1983 2809 2 IF PRESENT (name) then
1984 2810 2 BEGIN
1985 2811 2 LOCAL FOUND, VALUE;
1986 2812 2 FOUND = LOOKUP_SYMBOL (name, %NAME(name, '_DESC'), VALUE);
1987 2813 2 IF .FOUND XOR %NAME(name, '_SWITCH') THEN RETURN FALSE;
1988 2814 2 END%;
1989 2815 2
1990 2816 2
1991 2817 2
1992 2818 2
1993 2819 2
1994 2820 2 TEST THE RECORD --
1995 2821 2 Using tables and values built by PARSE_COMMAND test the input
1996 2822 2 record. Any time a record fails a selection test, or matches
1997 2823 2 an exclusion test, return immediately with an indicator to use
1998 2824 2 the secondary output stream. If all tests are passed, then return
1999 2825 2 TRUE.
2000 2826 2
2001 2827 2
2002 2828 2 Select_or_reject (account);
2003 2829 2
2004 2830 2 Select_or_reject (address);
2005 2831 2
2006 2832 2 Select_or_reject (entry);
2007 2833 2
2008 2834 2 Select_or_reject (ident);
2009 2835 2
2010 2836 2 Select_or_reject (image);
2011 2837 2
2012 2838 2 Select_or_reject (job);
2013 2839 2
```

```
2014 2840 2 Select_or_reject (node);
2015 2841 2
2016 2842 2 Select_or_reject (owner);
2017 2843 2
2018 2844 2 Select_or_reject (priority);
2019 2845 2
2020 2846 2 Select_or_reject (process);
2021 2847 2
2022 2848 2 Select_or_reject (queue);
2023 2849 2
2024 2850 2 Select_or_reject (remote_id);
2025 2851 2
2026 2852 2 Select_or_reject (status);
2027 2853 2
2028 2854 2 Select_or_reject (terminal);
2029 2855 2
2030 2856 2 Select_or_reject (type);
2031 2857 2
2032 2858 2 !+
2033 2859 2 ! Select or reject based on /UIC qualifier.
2034 2860 2 !-
2035 2861 2
2036 2862 2 begin
2037 2863 2
2038 2864 2 local
2039 2865 2     uicg, uicm, value;
2040 2866 2
2041 2867 2 uicg = lookup_symbol( uic_group, uic_group_desc, value );
2042 2868 2 uicm = lookup_symbol( uic_member, uic_member_desc, value );
2043 2869 2
2044 2870 2 !+
2045 2871 2 ! First, if both a group and member were specified, see if the record matches
2046 2872 2 ! BOTH. This is done so that a selection of /UIC=[1,3] won't reject records of
2047 2873 2 ! the form [*,3]. Then check for EITHER a match on group or member, but not
2048 2874 2 ! both.
2049 2875 2 !-
2050 2876 2
2051 2877 2 if present( uic_group ) and present( uic_member )
2052 2878 2 then
2053 2879 2     ( if ( .uicg xor .uic_group_switch ) and ( .uicm xor .uic_member_switch )
2054 2880 2     then
2055 2881 2         return false )
2056 2882 2 else
2057 2883 2     if present( uic_group )
2058 2884 2     then
2059 2885 2         ( if ( .uicg xor .uic_group_switch )
2060 2886 2         then
2061 2887 2             return false
2062 2888 2         )
2063 2889 2     else if present( uic_member )
2064 2890 2     then
2065 2891 2         if ( .uicm xor .uic_member_switch )
2066 2892 2         then
2067 2893 2             return false;
2068 2894 2
2069 2895 2 end;
2070 2896 2
```

```
2071 2897 2 Select_or_reject (user);
2072 2898 2
2073 2899 2
2074 2900 2
2075 2901 2
2076 2902 2 TEST DATES --
2077 2903 2
2078 2904 2 Check to see if this is the oldest or newest record
2079 2905 2 encountered. Update the date watermarks if so.
2080 2906 2
2081 2907 2
2082 2908 3 if COMPARE_QUAD (..date_desc_addr, lssu, first_date)
2083 2909 2 then MOVE_QUAD (..date_desc_addr, first_date);
2084 2910 2
2085 2911 3 if COMPARE_QUAD (..date_desc_addr, gtru, last_date)
2086 2912 2 then MOVE_QUAD (..date_desc_addr, last_date);
2087 2913 2
2088 2914 2
2089 2915 2
2090 2916 2
2091 2917 2 BEFORE_DATE and SINCE_DATE define a date range that the
2092 2918 2 record under test must satisfy. DATE_DESC_ADDR points to
2093 2919 2 one of three possible dates in the record to use in the
2094 2920 2 test. If the date pointed to does not fall inside the
2095 2921 2 range then return to the caller.
2096 2922 2
2097 2923 2
2098 2924 2 if (COMPARE_QUAD (..date_desc_addr, gtru, before_date)) or
2099 2925 3 (COMPARE_QUAD (..date_desc_addr, lssu, since_date))
2100 2926 2 then return false;
2101 2927 2
2102 2928 2
2103 2929 2
2104 2930 2 If code path falls through to here then the record has passed all
2105 2931 2 selection and rejection tests so count it as selected and return true.
2106 2932 2
2107 2933 2
2108 2934 2 selected = .selected + 1; ! Bump the selected record count
2109 2935 2 return true; ! Default is selected
2110 2936 2
2111 2937 1 END;
```

```
B= FIRST_DATE
B= LAST_DATE
B= BEFORE_DATE
B= SINCE_DATE
```

003C 00000 RECORD_SELECTED:

```
55 00000000G 00 9E 00002 .WORD Save R2,R3,R4,R5
54 00000000G 00 9E 00009 MOVAB LOOKUP_SYMBOL, R5
53 00000000' EF 9E 00010 MOVAB QUALIFIERS, R4
5E BC AE 9E 00017 MOVAB UIC_GROUP_SWITCH, R3
12 64 E9 0001B BLBC -68(TSP), SP
QUALIFIERS, 1$
```

: 2783

: 2828

			FE68	SE DD 0001E C3 9F 00020 7E D4 00024 03 FB 00026 C0 A3 C0 00029 50 E8 0002D 72 03 E1 00030 A4 04 AE 0F 00035 FE70 C3 9F 00038 0B DD 0003C 03 FB 0003E C4 A3 C0 00041 50 E8 00045 72 05 E1 00048 64 08 AE 9F 0004C FE90 C3 9F 0004F 05 DD 00053 03 FB 00055 C8 A3 C0 00058 50 E8 0005C 73 64 95 0005F 13 18 00061 0C AE 9F 00063 FEB8 C3 9F 00066 07 DD 0006A 03 FB 0006C CC A3 C0 0006F 50 E8 00073 74 01 A4 E9 00076 13 10 AE 9F 0007A FEC8 C3 9F 0007D 08 DD 00081 03 FB 00083 D0 A3 C0 00086 50 E8 0008A 74 01 E1 0008D A4 14 AE 9F 00092 FED0 C3 9F 00095 09 DD 00099 03 FB 0009B 50 A3 C0 0009E 74 05 E8 000A2 A4 04 E1 000A5 18 AE 9F 000AA FED8 C3 9F 000AD 0C DD 000B1 03 FB 000B3 D8 A3 C0 000B6 50 E8 000BA 74 05 E1 000BD A4 1C AE 9F 000C2 FEE0 C3 9F 000C5 0D DD 000C9 03 FB 000CB 50 A3 C0 000CE 74 05 E8 000D2 01 A4 95 000D5	PUSHL SP PUSHAB ACCOUNT_DESC CLRL -(SP) CALLS #3, LOOKUP SYMBOL ADDL2 ACCOUNT_SWITCH, R0 BLBS R0, 6\$ BBC #3, QUALIFIERS+1, 2\$ PUSHAB VALUE PUSHAB ADDRESS_DESC PUSHL #11 CALLS #3, LOOKUP SYMBOL ADDL2 ADDRESS_SWITCH, R0 BLBS R0, 8\$ BBC #5, QUALIFIERS, 3\$ PUSHAB VALUE PUSHAB ENTRY_DESC PUSHL #5 CALLS #3, LOOKUP SYMBOL ADDL2 ENTRY_SWITCH, R0 BLBS R0, 10\$ TSTB QUALIFIERS BGEQ 4\$ PUSHAB VALUE PUSHAB IDENT_DESC PUSHL #7 CALLS #3, LOOKUP SYMBOL ADDL2 IDENT_SWITCH, R0 BLBS R0, 12\$ BLBC QUALIFIERS+1, 5\$ PUSHAB VALUE PUSHAB IMAGE_DESC PUSHL #8 CALLS #3, LOOKUP SYMBOL ADDL2 IMAGE_SWITCH, R0 BLBS R0, 14\$ BBC #1, QUALIFIERS+1, 7\$ PUSHAB VALUE PUSHAB JOB_DESC PUSHL #9 CALLS #3, LOOKUP SYMBOL ADDL2 JOB_SWITCH, R0 BLBS R0, 16\$ BBC #4, QUALIFIERS+1, 9\$ PUSHAB VALUE PUSHAB NODE_DESC PUSHL #12 CALLS #3, LOOKUP SYMBOL ADDL2 NODE_SWITCH, R0 BLBS R0, 18\$ BBC #5, QUALIFIERS+1, 11\$ PUSHAB VALUE PUSHAB OWNER_DESC PUSHL #13 CALLS #3, LOOKUP SYMBOL ADDL2 OWNER_SWITCH, R0 BLBS R0, 20\$ TSTB QUALIFIERS+1	2830 2832 2834 2836 2838 2840 2842 2844
--	--	--	------	---	---	--

			13	18	000D8	BGEQ	13\$	
	20	AE	9F	000DA	PUSHAB	VALUE		
	FF00	C3	9F	000DD	PUSHAB	PRIORITY_DESC		
		0F	DD	000E1	PUSHL	#15		
	65	03	FB	000E3	CALLS	#3, LOOKUP_SYMBOL		
	50	A3	CO	000E6	ADDL2	PRIORITY_SWITCH, R0		
	74	50	E8	000EA	BLBS	R0, 22\$		
	13	02	A4	E9	000ED	13\$: QUALIFIERS+2, 15\$		2846
		24	AE	9F	000F1	13\$: BLBC		
		FF10	C3	9F	000F4	PUSHAB	VALUE	
			10	DD	000F8	PUSHAB	PROCESS_DESC	
			03	FB	000FA	PUSHL	#16	
	65	03	FB	000FA	CALLS	#3, LOOKUP_SYMBOL		
	50	A3	CO	000FD	ADDL2	PROCESS_SWITCH, R0		
	75	50	E8	00101	BLBS	R0, 24\$		
13	02	A4	01	E1	00104	14\$: BBC		2848
		28	AE	9F	00109	15\$: PUSHAB	VALUE	
		FF18	C3	9F	0010C	PUSHAB	QUEUE_DESC	
			11	DD	00110	PUSHL	#17	
	65	03	FB	00112	CALLS	#3, LOOKUP_SYMBOL		
	50	A3	CO	00115	ADDL2	QUEUE_SWITCH, R0		
	50	50	E8	00119	BLBS	R0, 24\$		
13	02	A4	03	E1	0011C	16\$: BBC		2850
		2C	AE	9F	00121	17\$: PUSHAB	VALUE	
		FF28	C3	9F	00124	PUSHAB	REMOTE_ID_DESC	
			13	DD	00128	PUSHL	#19	
	65	03	FB	0012A	CALLS	#3, LOOKUP_SYMBOL		
	50	A3	CO	0012D	ADDL2	REMOTE_ID_SWITCH, R0		
	45	50	E8	00131	BLBS	R0, 24\$		
13	02	A4	06	E1	00134	18\$: BBC		2852
		30	AE	9F	00139	19\$: PUSHAB	VALUE	
		FF38	C3	9F	0013C	PUSHAB	STATUS_DESC	
			16	DD	00140	PUSHL	#22	
	65	03	FB	00142	CALLS	#3, LOOKUP_SYMBOL		
	50	A3	CO	00145	ADDL2	STATUS_SWITCH, R0		
	76	50	E8	00149	BLBS	R0, 29\$		
13	03	A4	01	E1	0014C	20\$: BBC		2854
		34	AE	9F	00151	21\$: PUSHAB	VALUE	
		FF68	C3	9F	00154	PUSHAB	TERMINAL_DESC	
			19	DD	00158	PUSHL	#25	
	65	03	FB	0015A	CALLS	#3, LOOKUP_SYMBOL		
	50	A3	CO	0015D	ADDL2	TERMINAL_SWITCH, R0		
	74	50	E8	00161	BLBS	R0, 31\$		
13	03	A4	03	E1	00164	22\$: BBC		2856
		38	AE	9F	00169	23\$: PUSHAB	VALUE	
		FF70	C3	9F	0016C	PUSHAB	TYPE_DESC	
			18	DD	00170	PUSHL	#27	
	65	03	FB	00172	CALLS	#3, LOOKUP_SYMBOL		
	50	A3	CO	00175	ADDL2	TYPE_SWITCH, R0		
	5C	50	E8	00179	BLBS	R0, 31\$		
		3C	AE	9F	0017C	24\$: PUSHAB	VALUE	2867
		88	A3	9F	0017F	25\$: PUSHAB	UIC_GROUP_DESC	
			20	DD	00182	PUSHL	#32	
	65	03	FB	00184	CALLS	#3, LOOKUP_SYMBOL		
	52	50	DD	00187	MOVL	R0, UICG		
		3C	AE	9F	0018A	PUSHAB	VALUE	
		90	A3	9F	0018D	PUSHAB	UIC_MEMBER_DESC	
			21	DD	00190	PUSHL	#33	2868

50	04	A4	65	03	FB	00192	CALLS	#3, LOOKUP_SYMBOL	:	
			51	50	D0	00195	MOVL	R0, UICM	:	
			01	00	EF	00198	EXTZV	#0, #1, QUALIFIERS+4, R0	:	2877
			17	50	E9	0019E	BLBC	R0, 27\$	:	
		09	A4	01	E1	001A1	BBC	#1, QUALIFIERS+4, 26\$	:	
		50	52	63	C1	001A6	ADDL3	UIC_GROUP_SWITCH, UICG, R0	:	2879
			18	50	E9	001AA	BLBC	R0, 30\$	:	
				0E	11	001AD	BRB	28\$	:	
			06	50	E9	001AF	BLBC	R0, 27\$	:	2883
		50	52	63	C1	001B2	ADDL3	UIC_GROUP_SWITCH, UICG, R0	:	2885
				0A	11	001B6	BRB	29\$	:	
		08	A4	01	E1	001B8	BBC	#1, QUALIFIERS+4, 30\$	:	2889
		50	51	04	A3	001BD	ADDL3	UIC_MEMBER_SWITCH, UICM, R0	:	2891
			6C	50	E8	001C2	BLBS	R0, 40\$	:	
		12	64	04	E1	001C5	BBC	#4, QUALIFIERS, 32\$	:	2897
				40	AE	9F	PUSHAB	VALUE	:	
				A8	A3	9F	PUSHAB	USER_DESC	:	
			65	04	DD	001CF	PUSHL	#4	:	
			50	03	FB	001D1	CALLS	#3, LOOKUP_SYMBOL	:	
			56	08	A3	001D4	ADDL2	USER_SWITCH, R0	:	
			50	50	E8	001D8	BLBS	R0, 40\$	:	
		026C	C3	024C	D3	001DB	MOVL	@DATE_DESC_ADDR, R0	:	2908
				04	A0	001E0	CMPL	4(R0), B+4	:	
		0268	C3		05	12	BNEQ	33\$	:	
					60	D1	CMPL	(R0), B	:	
					05	1E	BGEQU	34\$	:	
		0268	C3		60	7D	MOVQ	(R0), FIRST_DATE	:	2909
		0264	C3	04	A0	001F4	CMPL	4(R0), B+4	:	2911
					05	12	BNEQ	35\$	:	
		0260	C3		60	D1	CMPL	(R0), B	:	
					05	1B	BLEQU	36\$	:	
		0260	C3		60	7D	MOVQ	(R0), LAST_DATE	:	2912
		0254	C3	04	A0	00208	CMPL	4(R0), B+4	:	2924
					09	12	BNEQ	37\$	:	
		0250	C3		60	D1	CMPL	(R0), B	:	
					04	1B	BLEQU	38\$	:	
					18	11	BRB	40\$	:	
					16	1A	BGTRU	40\$	:	
		025C	C3	04	A0	0021B	CMPL	4(R0), B+4	:	2925
					05	12	BNEQ	39\$	:	
		0258	C3		60	D1	CMPL	(R0), B	:	
					07	1F	BLSSU	40\$	:	
				0C	A3	0022A	INCL	SELECTED	:	2934
			50	01	D0	0022D	MOVL	#1, R0	:	2935
					04	00230	RET		:	
				50	D4	00231	CLRL	R0	:	2937
				04	00233	RET			:	

; Routine Size: 564 bytes, Routine Base: CODE + 1125

; 2112 2938 1

```
2114 2939 1 ROUTINE PARSE_COMMAND =
2115 2940 1
2116 2941 1 ---
2117 2942 1
2118 2943 1 Functional description
2119 2944 1
2120 2945 1 This routine is called from the main loop to process the command
2121 2946 1 line. Based on the presence of qualifiers and modifiers, routines
2122 2947 1 are called to parse the line.
2123 2948 1
2124 2949 1 Internal tables are defined, based on command line values, for later
2125 2950 1 processing.
2126 2951 1
2127 2952 1 Output parameters
2128 2953 1
2129 2954 1 Any errors encountered are RETURNed immediately.
2130 2955 1 TRUE is returned on a normal exit.
2131 2956 1
2132 2957 1 ----
2133 2958 1
2134 2959 2 BEGIN
2135 2960 2
2136 2961 2 UNDECLARE
2137 2962 2     uic_stb,
2138 2963 2     uic_ktb;
2139 2964 2
2140 2965 2 EXTERNAL
2141 2966 2     uic_stb,
2142 2967 2     uic_ktb;
2143 2968 2
2144 2969 2 LOCAL
2145 2970 2     first,                ! First time switch
2146 2971 2     temp;                 ! Temporary work area
2147 2972 2
2148 2973 2 OWN
2149 2974 2     desc: bblock [dsc$k_d_bln] ! Dynamic work descriptor
2150 2975 2     preset([dsc$b_c[lass] = dsc$k_class_d),
2151 2976 2
2152 2977 2
2153 2978 2
2154 2979 2 DEFINE KEYWORD TABLES ---
2155 2980 2 Define a keyword table for LIB$LOOKUP_KEY. Associate a number
2156 2981 2 with each keyword.
2157 2982 2
2158 2983 2
2159 2984 2
2160 P 2985 2 type keyword: $lib_key_table (      ! Keyword table -- record type
2161 P 2986 2     (CUSTOMER, ad ('CUSTOMER')),      ! -Customer record
2162 P 2987 2     (FILE, ad ('FILE')),              ! -File record
2163 P 2988 2     (IMAGE, ad ('IMAGE')),            ! -Image termination
2164 P 2989 2     (LOGFAIL, ad ('LOGFAIL')),        ! -Login failure
2165 P 2990 2     (PROCESS, ad ('PROCESS')),        ! -Process termination
2166 P 2991 2     (PRINT, ad ('PRINT')),            ! -Print job termination
2167 P 2992 2     (SETTIME, ad ('SETTIME')),        ! -System set time~r
2168 P 2993 2     (SYSINIT, ad ('SYSINIT')),        ! -System initialization
2169 P 2994 2     (UNKNOWN, ad ('UNKNOWN')),        ! -Unknown record types
2170 P 2995 2     (USER, ad ('USER')),              ! -User messages
```

```
2171      2996      2      ),
2172      2997      2
2173      2998      2      process keyword: $lib_key_table (      : Keyword table -- process type
2174      2999      2      (BATCH,      ad ('BATCH')),      : -Batch process
2175      3000      2      (DETACHED,      ad ('DETACHED')),      : -Detached process
2176      3001      2      (INTERACTIVE,      ad ('INTERACTIVE')),      : -Interactive process
2177      3002      2      (NETWORK,      ad ('NETWORK')),      : -Network process
2178      3003      2      (SUBPROCESS,      ad ('SUBPROCESS')),      : -Sub process
2179      3004      2      );
2180      3005      2
2181      3006      2
2182      3007      2
2183      3008      2
2184      3009      2      :
2185      3010      2      : PARSE COMMAND LINE MODIFIERS ---
2186      3011      2      : Call routines as necessary to parse command line modifiers and
2187      3012      2      : value lists. Exit immediately if any errors are returned by the
2188      3013      2      : parsing subroutines.
2189      3014      2      :
2190      3015      2
2191      3016      2      map_qualifiers ();      : Establish bitmask
2192      3017      2
2193      3018      2
2194      3019      2      :
2195      3020      2      : PARSE /ACCOUNT VALUE LIST --
2196      3021      2      :
2197      3022      2
2198      3023      2      First = true;
2199      3024      2      While GET_VALUE ('ACCOUNT', desc) do
2200      3025      3          BEGIN
2201      3026      4              if
2202      3027      4                  then
2203      3028      4                      BEGIN
2204      3029      4                          account_switch = strip_negator (desc);
2205      3030      4                          if .desc [desc$w_length] neq 0
2206      3031      4                          then perform (add_symbol (ACCOUNT, desc, 0));
2207      3032      4                          first = false;
2208      3033      4                      END
2209      3034      3                  else perform (add_symbol (ACCOUNT, desc, 0));
2210      3035      2              END;
2211      3036      2
2212      3037      2
2213      3038      2
2214      3039      2      :
2215      3040      2      : PARSE DATES --
2216      3041      2      :
2217      3042      2
2218      3043      2      perform (parse_dates ());      : Handle /DATE /BEFORE /SINCE
2219      3044      2
2220      3045      2
2221      3046      2
2222      3047      2
2223      3048      2      :
2224      3049      2      : PARSE /ADDRESS VALUE LIST --
2225      3050      2      :
2226      3051      2
2227      3052      2      First = true;
```

```
2228 3053 2 While GET VALUE ('ADDRESS', desc) do
2229 3054 3 BEGIN
2230 3055 3   if .first then (address_switch = strip_negator (desc); first = false);
2231 3056 3   if .desc [dsc$w_length] neq 0 then
2232 3057 4 BEGIN
2233 3058 5   if not (lib$cvd dtb (
2234 3059 5     .desc [dsc$w_length],
2235 3060 5     .desc [dsc$a_pointer],
2236 3061 5     .desc [dsc$a_pointer]))
2237 3062 4   then signal_return (msg$_syntax, 1, desc);
2238 3063 4   desc [dsc$w_length] = 4;
2239 3064 4   perform (add_symbol (address, desc, 0));
2240 3065 3 END;
2241 3066 2 END;
2242 3067 2
2243 3068 2
2244 3069 2
2245 3070 2
2246 3071 2 ! PARSE /ENTRY VALUE LIST --
2247 3072 2
2248 3073 2
2249 3074 2 First = true;
2250 3075 2 While GET VALUE ('ENTRY', desc) do
2251 3076 3 BEGIN
2252 3077 3   if .first then (entry_switch = strip_negator (desc); first = false);
2253 3078 3   if .desc [dsc$w_length] neq 0 then
2254 3079 4 BEGIN
2255 3080 5   if not (lib$cvd dtb (
2256 3081 5     .desc [dsc$w_length],
2257 3082 5     .desc [dsc$a_pointer],
2258 3083 5     .desc [dsc$a_pointer]))
2259 3084 4   then signal_return (msg$_syntax, 1, desc);
2260 3085 4   desc [dsc$w_length] = 4;
2261 3086 4   perform (add_symbol (entry, desc, 0));
2262 3087 3 END;
2263 3088 2 END;
2264 3089 2
2265 3090 2
2266 3091 2
2267 3092 2
2268 3093 2 ! PARSE /IDENTIFICATION VALUE LIST --
2269 3094 2
2270 3095 2
2271 3096 2 First = true;
2272 3097 2 While GET VALUE ('IDENT', desc) do
2273 3098 3 BEGIN
2274 3099 3   if .first then (ident_switch = strip_negator (desc); first = false);
2275 3100 3   if .desc [dsc$w_length] neq 0 then
2276 3101 4 BEGIN
2277 3102 5   if not (lib$cvd htb (
2278 3103 5     .desc [dsc$w_length],
2279 3104 5     .desc [dsc$a_pointer],
2280 3105 5     .desc [dsc$a_pointer]))
2281 3106 4   then signal_return (msg$_syntax, 1, desc);
2282 3107 4   desc [dsc$w_length] = 4;
2283 3108 4   perform (add_symbol (IDENT, desc, 0));
2284 3109 3 END;
```

```
2285 3110 2      END;
2286 3111 2
2287 3112 2
2288 3113 2
2289 3114 2  !
2290 3115 2  ! PARSE /IMAGE VALUE LIST --
2291 3116 2  !
2292 3117 2
2293 3118 2  First = true;
2294 3119 2  While GET VALUE ('IMAGE', desc) do
2295 3120 2      BEGIN
2296 3121 2          if .first
2297 3122 2          then BEGIN
2298 3123 2              image_switch = strip_negator (desc);
2299 3124 2              if .desc [dsc$w_length] neq 0
2300 3125 2              then perform (add_symbol (IMAGE, desc, 0));
2301 3126 2              first = 'false';
2302 3127 2          END
2303 3128 2          else perform (add_symbol (IMAGE, desc, 0));
2304 3129 2      END;
2305 3130 2
2306 3131 2
2307 3132 2
2308 3133 2  !
2309 3134 2  ! PARSE /JOB VALUE LIST --
2310 3135 2  !
2311 3136 2
2312 3137 2  First = true;
2313 3138 2  While GET VALUE ('JOB', desc) do
2314 3139 2      BEGIN
2315 3140 2          if .first
2316 3141 2          then BEGIN
2317 3142 2              job_switch = strip_negator (desc);
2318 3143 2              if .desc [dsc$w_length] neq 0
2319 3144 2              then perform (add_symbol (JOB, desc, 0));
2320 3145 2              first = false;
2321 3146 2          END
2322 3147 2          else perform (add_symbol (JOB, desc, 0));
2323 3148 2      END;
2324 3149 2
2325 3150 2
2326 3151 2  !
2327 3152 2  ! PARSE /NODE VALUE LIST --
2328 3153 2  !
2329 3154 2
2330 3155 2  First = true;
2331 3156 2  While GET VALUE ('NODE', desc) do
2332 3157 2      BEGIN
2333 3158 2          if .first
2334 3159 2          then BEGIN
2335 3160 2              node_switch = strip_negator (desc);
2336 3161 2              if .desc [dsc$w_length] neq 0
2337 3162 2              then perform (add_symbol (NODE, desc, 0));
2338 3163 2              first = false;
2339 3164 2          END
2340 3165 2          else perform (add_symbol (NODE, desc, 0));
2341 3166 2      END;
```

```

3167 2
3168 2
3169 2
3170 2
3171 2
3172 2
3173 2
3174 2
3175 2
3176 2
3177 2
3178 2
3179 2
3180 2
3181 2
3182 2
3183 2
3184 2
3185 2
3186 2
3187 2
3188 2
3189 2
3190 2
3191 2
3192 2
3193 2
3194 2
3195 2
3196 2
3197 2
3198 2
3199 2
3200 2
3201 2
3202 2
3203 2
3204 2
3205 2
3206 2
3207 2
3208 2
3209 2
3210 2
3211 2
3212 2
3213 2
3214 2
3215 2
3216 2
3217 2
3218 2
3219 2
3220 2
3221 2
3222 2
3223 2
3224 2
3225 2
3226 2
3227 2
3228 2
3229 2
3230 2
3231 2
3232 2
3233 2
3234 2
3235 2
3236 2
3237 2
3238 2
3239 2
3240 2
3241 2
3242 2
3243 2
3244 2
3245 2
3246 2
3247 2
3248 2
3249 2
3250 2
3251 2
3252 2
3253 2
3254 2
3255 2
3256 2
3257 2
3258 2
3259 2
3260 2
3261 2
3262 2
3263 2
3264 2
3265 2
3266 2
3267 2
3268 2
3269 2
3270 2
3271 2
3272 2
3273 2
3274 2
3275 2
3276 2
3277 2
3278 2
3279 2
3280 2
3281 2
3282 2
3283 2
3284 2
3285 2
3286 2
3287 2
3288 2
3289 2
3290 2
3291 2
3292 2
3293 2
3294 2
3295 2
3296 2
3297 2
3298 2
3299 2
3300 2
3301 2
3302 2
3303 2
3304 2
3305 2
3306 2
3307 2
3308 2
3309 2
3310 2
3311 2
3312 2
3313 2
3314 2
3315 2
3316 2
3317 2
3318 2
3319 2
3320 2
3321 2
3322 2
3323 2
3324 2
3325 2
3326 2
3327 2
3328 2
3329 2
3330 2
3331 2
3332 2
3333 2
3334 2
3335 2
3336 2
3337 2
3338 2
3339 2
3340 2
3341 2
3342 2
3343 2
3344 2
3345 2
3346 2
3347 2
3348 2
3349 2
3350 2
3351 2
3352 2
3353 2
3354 2
3355 2
3356 2
3357 2
3358 2
3359 2
3360 2
3361 2
3362 2
3363 2
3364 2
3365 2
3366 2
3367 2
3368 2
3369 2
3370 2
3371 2
3372 2
3373 2
3374 2
3375 2
3376 2
3377 2
3378 2
3379 2
3380 2
3381 2
3382 2
3383 2
3384 2
3385 2
3386 2
3387 2
3388 2
3389 2
3390 2
3391 2
3392 2
3393 2
3394 2
3395 2
3396 2
3397 2
3398 2
3399 2
3400 2
3401 2
3402 2
3403 2
3404 2
3405 2
3406 2
3407 2
3408 2
3409 2
3410 2
3411 2
3412 2
3413 2
3414 2
3415 2
3416 2
3417 2
3418 2
3419 2
3420 2
3421 2
3422 2
3423 2
3424 2
3425 2
3426 2
3427 2
3428 2
3429 2
3430 2
3431 2
3432 2
3433 2
3434 2
3435 2
3436 2
3437 2
3438 2
3439 2
3440 2
3441 2
3442 2
3443 2
3444 2
3445 2
3446 2
3447 2
3448 2
3449 2
3450 2
3451 2
3452 2
3453 2
3454 2
3455 2
3456 2
3457 2
3458 2
3459 2
3460 2
3461 2
3462 2
3463 2
3464 2
3465 2
3466 2
3467 2
3468 2
3469 2
3470 2
3471 2
3472 2
3473 2
3474 2
3475 2
3476 2
3477 2
3478 2
3479 2
3480 2
3481 2
3482 2
3483 2
3484 2
3485 2
3486 2
3487 2
3488 2
3489 2
3490 2
3491 2
3492 2
3493 2
3494 2
3495 2
3496 2
3497 2
3498 2
3499 2
3500 2
3501 2
3502 2
3503 2
3504 2
3505 2
3506 2
3507 2
3508 2
3509 2
3510 2
3511 2
3512 2
3513 2
3514 2
3515 2
3516 2
3517 2
3518 2
3519 2
3520 2
3521 2
3522 2
3523 2
3524 2
3525 2
3526 2
3527 2
3528 2
3529 2
3530 2
3531 2
3532 2
3533 2
3534 2
3535 2
3536 2
3537 2
3538 2
3539 2
3540 2
3541 2
3542 2
3543 2
3544 2
3545 2
3546 2
3547 2
3548 2
3549 2
3550 2
3551 2
3552 2
3553 2
3554 2
3555 2
3556 2
3557 2
3558 2
3559 2
3560 2
3561 2
3562 2
3563 2
3564 2
3565 2
3566 2
3567 2
3568 2
3569 2
3570 2
3571 2
3572 2
3573 2
3574 2
3575 2
3576 2
3577 2
3578 2
3579 2
3580 2
3581 2
3582 2
3583 2
3584 2
3585 2
3586 2
3587 2
3588 2
3589 2
3590 2
3591 2
3592 2
3593 2
3594 2
3595 2
3596 2
3597 2
3598 2
3599 2
3600 2
3601 2
3602 2
3603 2
3604 2
3605 2
3606 2
3607 2
3608 2
3609 2
3610 2
3611 2
3612 2
3613 2
3614 2
3615 2
3616 2
3617 2
3618 2
3619 2
3620 2
3621 2
3622 2
3623 2
3624 2
3625 2
3626 2
3627 2
3628 2
3629 2
3630 2
3631 2
3632 2
3633 2
3634 2
3635 2
3636 2
3637 2
3638 2
3639 2
3640 2
3641 2
3642 2
3643 2
3644 2
3645 2
3646 2
3647 2
3648 2
3649 2
3650 2
3651 2
3652 2
3653 2
3654 2
3655 2
3656 2
3657 2
3658 2
3659 2
3660 2
3661 2
3662 2
3663 2
3664 2
3665 2
3666 2
3667 2
3668 2
3669 2
3670 2
3671 2
3672 2
3673 2
3674 2
3675 2
3676 2
3677 2
3678 2
3679 2
3680 2
3681 2
3682 2
3683 2
3684 2
3685 2
3686 2
3687 2
3688 2
3689 2
3690 2
3691 2
3692 2
3693 2
3694 2
3695 2
3696 2
3697 2
3698 2
3699 2
3700 2
3701 2
3702 2
3703 2
3704 2
3705 2
3706 2
3707 2
3708 2
3709 2
3710 2
3711 2
3712 2
3713 2
3714 2
3715 2
3716 2
3717 2
3718 2
3719 2
3720 2
3721 2
3722 2
3723 2
3724 2
3725 2
3726 2
3727 2
3728 2
3729 2
3730 2
3731 2
3732 2
3733 2
3734 2
3735 2
3736 2
3737 2
3738 2
3739 2
3740 2
3741 2
3742 2
3743 2
3744 2
3745 2
3746 2
3747 2
3748 2
3749 2
3750 2
3751 
```

```

2399 3224 4      then BEGIN
2400 3225 4      process_switch = strip_negator (desc);
2401 3226 4      if .desc [dsc$w_length] neq 0
2402 3227 5      then BEGIN
2403 3228 5          LOOKUP (desc, process_keyword, process_value);
2404 3229 5          perform (ADD_SYMBOL (PROCESS, .process_value, 0));
2405 3230 4          END;
2406 3231 4          first = false;
2407 3232 4          END
2408 3233 4      else BEGIN
2409 3234 4          LOOKUP (desc, process_keyword, process_value);
2410 3235 4          perform (ADD_SYMBOL (PROCESS, .process_value, 0));
2411 3236 3          END;
2412 3237 2      END;
2413 3238 2
2414 3239 2
2415 3240 2
2416 3241 2
2417 3242 2  !
2418 3243 2  !PARSE /QUEUE VALUE LIST --
2419 3244 2  !
2420 3245 2
2421 3246 2  First = true;
2422 3247 2  While GET_VALUE ('QUEUE', desc) do
2423 3248 3      BEGIN
2424 3249 3          if .first
2425 3250 4          then BEGIN
2426 3251 4              queue_switch = strip_negator (desc);
2427 3252 4              if .desc [dsc$w_length] neq 0
2428 3253 4              then perform (add_symbol (QUEUE, desc, 0));
2429 3254 4              first = false;
2430 3255 4              END
2431 3256 3          else perform (add_symbol (QUEUE, desc, 0));
2432 3257 2          END;
2433 3258 2
2434 3259 2
2435 3260 2
2436 3261 2  !
2437 3262 2  !PARSE /REMOTE_ID VALUE LIST --
2438 3263 2  !
2439 3264 2
2440 3265 2  First = true;
2441 3266 2  While GET_VALUE ('REMOTE_ID', desc) do
2442 3267 3      BEGIN
2443 3268 3          if .first
2444 3269 4          then BEGIN
2445 3270 4              remote_id_switch = strip_negator (desc);
2446 3271 4              if .desc [dsc$w_length] neq 0
2447 3272 4              then perform (add_symbol (REMOTE_ID, desc, 0));
2448 3273 4              first = false;
2449 3274 4              END
2450 3275 3          else perform (add_symbol (REMOTE_ID, desc, 0));
2451 3276 2          END;
2452 3277 2
2453 3278 2
2454 3279 2
2455 3280 2  !
```

```
2456 3281 2 !PARSE /REPORT, /SORT, /SUMMARY --
2457 3282 3 !
2458 3283 3 !
2459 3284 3 Perform (parse_keys ());
2460 3285 3
2461 3286 3 !
2462 3287 3 !
2463 3288 3 !PARSE /STATUS VALUE LIST --
2464 3289 3 !
2465 3290 3
2466 3291 3 First = true;
2467 3292 3 While GET VALUE ('STATUS', desc) do
2468 3293 3 BEGIN
2469 3294 3 if .first then (status_switch = strip_negator (desc); first = false);
2470 3295 3 if .desc [dsc$w_length] neq 0 then
2471 3296 3 BEGIN
2472 3297 3 LOCAL status_status;
2473 3298 3 If not (lib$cvl_hb (
2474 3299 3 .desc [dsc$w_length],
2475 3300 3 .desc [dsc$a_pointer],
2476 3301 3 .desc [dsc$a_pointer]))
2477 3302 3 then signal_return (msg$_syntax, 1, desc);
2478 3303 3 desc [dsc$w_length] = 4;
2479 3304 3 status_status = add_symbol (STATUS, desc, 0);
2480 3305 3 if not .status_status then return .status_status;
2481 3306 3 END;
2482 3307 3 END;
2483 3308 3
2484 3309 3
2485 3310 3
2486 3311 3 !
2487 3312 3 !PARSE /TERMINAL VALUE LIST --
2488 3313 3 !
2489 3314 3 !
2490 3315 3 First = true;
2491 3316 3 While GET VALUE ('TERMINAL', desc) do
2492 3317 3 BEGIN
2493 3318 3 if
2494 3319 3 then BEGIN
2495 3320 3 terminal_switch = strip_negator (desc);
2496 3321 3 if .desc [dsc$w_length] neq 0
2497 3322 3 then perform (add_symbol (TERMINAL, desc, 0));
2498 3323 3 first = false;
2499 3324 3 END
2500 3325 3 else perform (add_symbol (TERMINAL, desc, 0));
2501 3326 3 END;
2502 3327 3
2503 3328 3
2504 3329 3
2505 3330 3 !
2506 3331 3 !PARSE /TYPE VALUE LIST --
2507 3332 3 !
2508 3333 3 !
2509 3334 3 First = true;
2510 3335 3 While GET VALUE ('TYPE', desc) do
2511 3336 3 BEGIN
2512 3337 3 LOCAL type_value;
```



```
: 2570      3395 6      qualifiers [uic_group] = 1;
: 2571      3396 6      uic_group_switch = .uic_switch;
: 2572      3397 6      add_symbol (UIC_GROUP, group_desc, 0);
: 2573      3398 5      END;
: 2574      3399 5      if
: 2575      3400 6      then
: 2576      3401 6      BEGIN
: 2577      3402 6      EXTERNAL qualifiers: BITVECTOR [];
: 2578      3403 6      qualifiers [uic_member] = 1;
: 2579      3404 6      uic_member_switch = .uic_switch;
: 2580      3405 5      add_symbol (UIC_MEMBER, member_desc, 0);
: 2581      3406 4      END;
: 2582      3407 3      END;
: 2583      3408 2      END;
: 2584      3409 2      !
: 2585      3410 2      ! PARSE /USER VALUE LIST --
: 2586      3411 2      !
: 2587      3412 2      !
: 2588      3413 2      First = true;
: 2589      3414 2      While GET_VALUE ('USER', desc) do
: 2590      3415 2      BEGIN
: 2591      3416 3      if
: 2592      3417 3      then
: 2593      3418 4      .first
: 2594      3419 4      BEGIN
: 2595      3420 4      user_switch = strip_negator (desc);
: 2596      3421 4      if .desc [desc$w_length] neg 0
: 2597      3422 4      then perform (add_symbol (USER, desc, 0));
: 2598      3423 4      first = false;
: 2599      3424 3      END
: 2600      3425 2      else
: 2601      3426 2      perform (add_symbol (USER, desc, 0));
: 2602      3427 2      END;
: 2603      3428 2
: 2604      3429 2      return true;
: 2605      3430 2
: 2606      3431 1      END;
```

```
00 00 00 52 45 4D 4F 54 53 55 43 08 01359 .BLKB 3
52 45 4D 4F 54 53 55 43 0135C P.ACF: .ASCII <8>\CUSTOMER\<0><0><0>
00000008 01368 P.ACH: .ASCII \CUSTOMER\
00000000 01370 P.ACG: .LONG 8
00 00 00 45 4C 49 46 04 01374 .ADDRESS P.ACH
45 4C 49 46 01378 P.ACI: .ASCII <4>\FILE\<0><0><0>
00000004 01380 P.ACK: .ASCII \FILE\
00000000 01384 P.ACJ: .LONG 4
00 00 45 47 41 4D 49 05 01388 .ADDRESS P.ACK
00 00 00 45 47 41 4D 49 0138C P.ACL: .ASCII <5>\IMAGE\<0><0>
00000005 01394 P.ACN: .ASCII \IMAGE\<0><0><0>
00000000 0139C P.ACM: .LONG 5
4C 49 41 46 47 4F 4C 07 013A0 .ADDRESS P.ACN
00 4C 49 41 46 47 4F 4C 013A4 P.ACO: .ASCII <7>\LOGFAIL\
00000007 013AC P.ACQ: .ASCII \LOGFAIL\<0>
00000000 013B4 P.ACP: .LONG 7
00000000 013B8 .ADDRESS P.ACQ
```

```
53 53 45 43 4F 52 50 07 013BC P.ACR: .ASCII <7>\PROCESS\  
00 53 53 45 43 4F 52 50 013C4 P.ACT: .ASCII \PROCESS\<0>  
00000007 013CC P.ACS: .LONG 7  
00000000 013D0 .ADDRESS P.ACT  
00 00 54 4E 49 52 50 05 013D4 P.ACU: .ASCII <5>\PRINT\<0><0>  
00 00 00 54 4E 49 52 50 013DC P.ACW: .ASCII \PRINT\<0><0><0>  
00000005 013E4 P.ACV: .LONG 5  
00000000 013E8 .ADDRESS P.ACW  
45 4D 49 54 54 45 53 07 013EC P.ACX: .ASCII <7>\SETTIME\  
00 45 4D 49 54 54 45 53 013F4 P.ACZ: .ASCII \SETTIME\<0>  
00000007 013FC P.ACY: .LONG 7  
00000000 01400 .ADDRESS P.ACZ  
54 49 4E 49 53 59 53 07 01404 P.ADA: .ASCII <7>\SYSINIT\  
00 54 49 4E 49 53 59 53 0140C P.ADC: .ASCII \SYSINIT\<0>  
00000007 01414 P.ADB: .LONG 7  
00000000 01418 .ADDRESS P.ADC  
4E 57 4F 4E 4B 4E 55 07 0141C P.ADD: .ASCII <7>\UNKNOWN\  
00 4E 57 4F 4E 4B 4E 55 01424 P.ADF: .ASCII \UNKNOWN\<0>  
00000007 0142C P.ADE: .LONG 7  
00000000 01430 .ADDRESS P.ADF  
00 00 00 52 45 53 55 04 01434 P.ADG: .ASCII <4>\USER\<0><0><0>  
00 00 00 52 45 53 55 0143C P.ADI: .ASCII \USER\  
00000004 01440 P.ADH: .LONG 4  
00000000 01444 .ADDRESS P.ADI  
00 00 48 43 54 41 42 05 01448 P.ADJ: .ASCII <5>\BATCH\<0><0>  
00 00 00 48 43 54 41 42 01450 P.ADL: .ASCII \BATCH\<0><0><0>  
00000005 01458 P.ADK: .LONG 5  
00000000 0145C .ADDRESS P.ADL  
00 00 00 44 45 48 43 41 54 45 44 08 01460 P.ADM: .ASCII <8>\DETACHED\<0><0><0>  
44 45 48 43 41 54 45 44 0146C P.ADO: .ASCII \DETACHED\  
00000008 01474 P.ADN: .LONG 8  
00000000 01478 .ADDRESS P.ADO  
45 56 49 54 43 41 52 45 54 4E 49 0B 0147C P.ADP: .ASCII <11>\INTERACTIVE\  
00 45 56 49 54 43 41 52 45 54 4E 49 01488 P.ADR: .ASCII \INTERACTIVE\<0>  
0000000B 01494 P.ADQ: .LONG 11  
00000000 01498 .ADDRESS P.ADR  
4B 52 4F 57 54 45 4E 07 0149C P.ADS: .ASCII <7>\NETWORK\  
00 4B 52 4F 57 54 45 4E 014A4 P.ADU: .ASCII \NETWORK\<0>  
00000007 014AC P.ADT: .LONG 7  
00000000 014B0 .ADDRESS P.ADU  
00 53 53 45 43 4F 52 50 42 55 53 0A 014B4 P.ADV: .ASCII <10>\SUBPROCESS\<0>  
00 00 53 53 45 43 4F 52 50 42 55 53 014C0 P.ADX: .ASCII \SUBPROCESS\<0><0>  
0000000A 014CC P.ADW: .LONG 10  
00000000 014D0 .ADDRESS P.ADX  
00 54 4E 55 4F 43 43 41 014D4 P.ADZ: .ASCII \ACCOUNT\<0>  
00000007 014DC P.ADY: .LONG 7  
00000000 014E0 .ADDRESS P.ADZ  
00 53 53 45 52 44 44 41 014E4 P.AEB: .ASCII \ADDRESS\<0>  
00000007 014EC P.AEA: .LONG 7  
00000000 014F0 .ADDRESS P.AEB  
00 00 0C 59 52 54 4E 45 014F4 P.AED: .ASCII \ENTRY\<0><0><0>  
00000005 014FC P.AEC: .LONG 5  
00000000 01500 .ADDRESS P.AED  
00 00 00 54 4E 45 44 49 01504 P.AEF: .ASCII \IDENT\<0><0><0>  
00000005 0150C P.AEE: .LONG 5  
00000000 01510 .ADDRESS P.AEF  
00 00 00 45 47 41 4D 49 01514 P.AEH: .ASCII \IMAGE\<0><0><0>
```

```
00000005 0151C P.AEG: .LONG 5
00000000' 01520 .ADDRESS P.AEH
00 42 4F 4A 01524 P.AEJ: .ASCII \JOB\<0>
00000003' 01528 P.AEI: .LONG 3
00000000' 0152C .ADDRESS P.AEJ
45 44 4F 4E 01530 P.AEL: .ASCII \NODE\
00000004' 01534 P.AEK: .LONG 4
00000000' 01538 .ADDRESS P.AEL
00 00 00 52 45 4E 57 4F 0153C P.AEN: .ASCII \OWNER\<0><0><0>
00000005' 01544 P.AEM: .LONG 5
00000000' 01548 .ADDRESS P.AEN
59 54 49 52 4F 49 52 50 0154C P.AEP: .ASCII \PRIORITY\
00000008' 01554 P.AEO: .LONG 8
00000000' 01558 .ADDRESS P.AEP
00 53 53 45 43 4F 52 50 0155C P.AER: .ASCII \PROCESS\<0>
00000007' 01564 P.AEQ: .LONG 7
00000000' 01568 .ADDRESS P.AER
00 00 00 45 55 45 55 51 0156C P.AET: .ASCII \QUEUE\<0><0><0>
00000005' 01574 P.AES: .LONG 5
00000000' 01578 .ADDRESS P.AET
70 00 00 44 49 5F 45 54 4F 4D 45 52 0157C P.AEV: .ASCII \REMOTE_ID\<0><0><0>
00000009' 01588 P.AEU: .LONG 9
00000000' 0158C .ADDRESS P.AEV
00 00 53 55 54 41 54 53 01590 P.AEX: .ASCII \STATUS\<0><0>
00000006' 01598 P.AEW: .LONG 6
00000000' 0159C .ADDRESS P.AEX
4C 41 4E 49 4D 52 45 54 015A0 P.AEZ: .ASCII \TERMINAL\
00000008' 015A8 P.AEY: .LONG 8
00000000' 015AC .ADDRESS P.AEZ
45 50 59 54 015B0 P.AFB: .ASCII \TYPE\
00000004' 015B4 P.AFA: .LONG 4
00000000' 015B8 .ADDRESS P.AFB
00 43 49 55 015BC P.AFD: .ASCII \UIC\<0>
00000003' 015C0 P.AFC: .LONG 3
00000000' 015C4 .ADDRESS P.AFD
52 45 53 55 015C8 P.AFF: .ASCII \USER\
00000004' 015CC P.AFE: .LONG 4
00000000' 015D0 .ADDRESS P.AFF

.PSECT $OWNS,NOEXE,2

00# 00010 DESC: .BYTE 0[3]
02 00013 .BYTE 2
00014 .BLKB 4
00000014 00018 TYPE_KEYWORD:
.LONG 20
00000000' 0001C .ADDRESS P.ACF, P.ACG, P.ACI, P.ACJ, P.ACL, -
00000000' 00034 P.ACM, P.ACO, P.ACP, P.ACR, P.ACS, P.ACU, -
00000000' 0004C P.ACV, P.ACX, P.ACY, P.ADA, P.ADB, P.ADD, -
00000000' 00064 P.ADE, P.ADG, P.ADH
0000000A 0006C PROCESS_KEYWORD:
.LONG 10
00000000' 00070 .ADDRESS P.ADJ, P.ADK, P.ADM, P.ADN, P.ADP, -
00000000' 00088 P.ADQ, P.ADS, P.ADT, P.ADV, P.ADW
00000002 00098 GROUP_DESC:
.LONG 2
00000000' 0009C .ADDRESS GROUP
```

```
00000002 000A0 MEMBER_DESC:
                                .LONG 2
00000003 00000000' 000A4 .ADDRESS MEMBER
00000008 000A8 TPARSE_BLOCK:
                                .LONG 8 3
                                .BLKB 28

.PSECT CODE,NOWRT,2

OFFC 00000 PARSE_COMMAND:
.WORD Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11
5B 00000000G 00 9E 00002 MOVAB LIB$CVT-HTB, R11
5A 00000000G 00 9E 00009 MOVAB LIB$CVT-DTB, R10
59 00000000G 00 9E 00010 MOVAB LIB$LOOKUP_KEY, R9
58 FEED CF 9E 00017 MOVAB P.ADY, R8
57 00000000G 00 9E 0001C MOVAB STRIP_NEGATOR, R7
56 00000000G 00 9E 00023 MOVAB CLISGET_VALUE, R6
55 00000000G 00 9E 0002A MOVAB ADD_SYMBOL, R5
54 00000000' EF 9E 00031 MOVAB UIC_SWITCH, R4
53 0000' CF 9E 00038 MOVAB DESC, R3
5E 08 C2 0003D SUBL2 #8, SP
00000000G 00 00 FB 00040 CALLS #0, MAP_QUALIFIERS
52 01 D0 00047 MOVL #1, FIRST
53 DD 0004A 1$: PUSHL R3
58 DD 0004C PUSHL R8
66 02 FB 0004E CALLS #2, CLISGET_VALUE
2D 50 E9 00051 BLBC R0, 4$
1D 52 E9 00054 BLBC FIRST, 3$
53 DD 00057 PUSHL R3
67 01 FB 00059 CALLS #1, STRIP_NEGATOR
C4 A4 50 D0 0005C MOVL R0, ACCOUNT_SWITCH
63 B5 00060 TSTW DESC
0C 13 00062 BEQL 2$
7E D4 00064 CLRL -(SP)
53 DD 00066 PUSHL R3
7E D4 00068 CLRL -(SP)
65 03 FB 0006A CALLS #3, ADD_SYMBOL
16 50 E9 0006D BLBC STATUS, -5$
52 D4 00070 2$: CLRL FIRST
D6 11 00072 BRB 1$
7E D4 00074 3$: CLRL -(SP)
53 DD 00076 PUSHL R3
7E D4 00078 CLRL -(SP)
65 03 FB 0007A CALLS #3, ADD_SYMBOL
CA 50 E8 0007D BLBS STATUS, -1$
04 00080 RET
0000V CF 00 FB 00081 4$: CALLS #0, PARSE_DATES
01 50 E8 00086 5$: BLBS STATUS, 6$
04 00089 RET
52 01 D0 0008A 6$: MOVL #1, FIRST
10 53 DD 0008D 7$: PUSHL R3
A8 9F 0008F PUSHAB P.AEA
66 02 FB 00092 CALLS #2, CLISGET_VALUE
31 50 E9 00095 BLBC R0, 9$
0B 52 E9 00098 BLBC FIRST, 8$
```

		53	DD	0009B	PUSHL	R3		
	67	01	FB	0009D	CALLS	#1, STRIP_NEGATOR		
C8	A4	50	DD	000A0	MOVL	R0, ADDRESS_SWITCH		
		52	D4	000A4	CLRL	FIRST		
	50	63	3C	000A6	8\$: MOVZWL	DESC, R0		3056
		E2	13	000A9	BEQL	7\$		
		04	A3	DD	000AB	PUSHL	DESC+4	3061
		04	A3	DD	000AE	PUSHL	DESC+4	3060
		50	DD	000B1	PUSHL	R0		3059
	6A	03	FB	000B3	CALLS	#3, LIB\$CVT_DTB		
	7B	50	E9	000B6	BLBC	R0, 15\$		
	63	04	B0	000B9	MOVW	#4, DESC		3063
		7E	D4	000BC	CLRL	-(SP)		3064
		53	DD	000BE	PUSHL	R3		
		0B	DD	000C0	PUSHL	#11		
	65	03	FB	000C2	CALLS	#3, ADD_SYMBOL		
	C5	50	E8	000C5	BLBS	STATUS, -7\$		
		04	000C8	RET				
	52	01	DD	000C9	9\$: MOVL	#1, FIRST		3074
		53	DD	000CC	10\$: PUSHL	R3		3075
		A8	9F	000CE	PUSHAB	P.AEC		
	66	02	FB	000D1	CALLS	#2, CLISGET_VALUE		
	31	50	E9	000D4	BLBC	R0, 12\$		
	0B	52	E9	000D7	BLBC	FIRST, 11\$		3077
		53	DD	000DA	PUSHL	R3		
	67	01	FB	000DC	CALLS	#1, STRIP_NEGATOR		
CC	A4	50	DD	000DF	MOVL	R0, ENTRY_SWITCH		
		52	D4	000E3	CLRL	FIRST		
	50	63	3C	000E5	11\$: MOVZWL	DESC, R0		3078
		E2	13	000E8	BEQL	10\$		
		04	A3	DD	000EA	PUSHL	DESC+4	3083
		04	A3	DD	000ED	PUSHL	DESC+4	3082
		50	DD	000F0	PUSHL	R0		3081
	6A	03	FB	000F2	CALLS	#3, LIB\$CVT_DTB		
	3C	50	E9	000F5	BLBC	R0, 15\$		
	63	04	B0	000F8	MOVW	#4, DESC		3085
		7E	D4	000FB	CLRL	-(SP)		3086
		53	DD	000FD	PUSHL	R3		
		05	DD	000FF	PUSHL	#5		
	65	03	FB	00101	CALLS	#3, ADD_SYMBOL		
	C5	50	E8	00104	BLBS	STATUS, -10\$		
		04	00107	RET				
	52	01	DD	00108	12\$: MOVL	#1, FIRST		3096
		53	DD	0010B	13\$: PUSHL	R3		3097
		A8	9F	0010D	PUSHAB	P.AEE		
	66	02	FB	00110	CALLS	#2, CLISGET_VALUE		
	34	50	E9	00113	BLBC	R0, 17\$		
	0B	52	E9	00116	BLBC	FIRST, 14\$		3099
		53	DD	00119	PUSHL	R3		
	67	01	FB	0011B	CALLS	#1, STRIP_NEGATOR		
DO	A4	50	DD	0011E	MOVL	R0, IDENT_SWITCH		
		52	D4	00122	CLRL	FIRST		
	50	63	3C	00124	14\$: MOVZWL	DESC, R0		3100
		E2	13	00127	BEQL	13\$		
		04	A3	DD	00129	PUSHL	DESC+4	3105
		04	A3	DD	0012C	PUSHL	DESC+4	3104
		50	DD	0012F	PUSHL	R0		3103

68	03	FB	00131	CALLS	#3, LIB\$CVT_MTB	
03	50	E8	00134	BLBS	R0, 16\$	
	035A	31	00137	BRW	72\$	
63	04	B0	0013A	MOVW	#4, DESC	3107
	7E	D4	0013D	CLRL	-(SP)	3108
	53	DD	0013F	PUSHL	R3	
	07	DD	00141	PUSHL	#7	
65	03	FB	00143	CALLS	#3, ADD_SYMBOL	
C2	50	E8	00146	BLBS	STATUS, -13\$	
		04	00149	RET		
52	01	DD	0014A	MOVL	#1, FIRST	3118
	53	DD	0014D	PUSHL	R3	3119
	40	A8	9F	PUSHAB	P.AEG	
66	02	FB	00152	CALLS	#2, CLISGET_VALUE	
2D	50	E9	00155	BLBC	R0, 21\$	
1D	52	E9	00158	BLBC	FIRST, 20\$	3121
	53	DD	0015B	PUSHL	R3	3123
67	01	FB	0015D	CALLS	#1, STRIP_NEGATOR	
D4	50	DD	00160	MOVL	R0, IMAGE_SWITCH	
A4	63	B5	00164	TSTW	DESC	3124
	0C	13	00166	BEQL	19\$	
	7E	D4	00168	CLRL	-(SP)	3125
	53	DD	0016A	PUSHL	R3	
	08	DD	0016C	PUSHL	#8	
65	03	FB	0016E	CALLS	#3, ADD_SYMBOL	
73	50	E9	00171	BLBC	STATUS, -27\$	
	52	D4	00174	CLRL	FIRST	3126
	D5	11	00176	BRB	18\$	3121
	7E	D4	00178	CLRL	-(SP)	3128
	53	DD	0017A	PUSHL	R3	
	08	DD	0017C	PUSHL	#8	
65	03	FB	0017E	CALLS	#3, ADD_SYMBOL	
C9	50	E8	00181	BLBS	STATUS, -18\$	
		04	00184	RET		
52	01	DD	00185	MOVL	#1, FIRST	3137
	53	DD	00188	PUSHL	R3	3138
	40	A8	9F	PUSHAB	P.AEI	
66	02	FB	0018D	CALLS	#2, CLISGET_VALUE	
2D	50	E9	00190	BLBC	R0, 25\$	
1D	52	E9	00193	BLBC	FIRST, 24\$	3140
	53	DD	00196	PUSHL	R3	3142
67	01	FB	00198	CALLS	#1, STRIP_NEGATOR	
D8	50	DD	0019B	MOVL	R0, JOB_SWITCH	
A4	63	B5	0019F	TSTW	DESC	3143
	0C	13	001A1	BEQL	23\$	
	7E	D4	001A3	CLRL	-(SP)	3144
	53	DD	001A5	PUSHL	R3	
	09	DD	001A7	PUSHL	#9	
65	03	FB	001A9	CALLS	#3, ADD_SYMBOL	
38	50	E9	001AC	BLBC	STATUS, -27\$	
	52	D4	001AF	CLRL	FIRST	3145
	D5	11	001B1	BRB	22\$	3140
	7E	D4	001B3	CLRL	-(SP)	3147
	53	DD	001B5	PUSHL	R3	
	09	DD	001B7	PUSHL	#9	
65	03	FB	001B9	CALLS	#3, ADD_SYMBOL	
C9	50	E8	001BC	BLBS	STATUS, -22\$	

			04 001BF	RET		
52		01	D0 001C0	25\$: MOVL	#1, FIRST	3155
		53	DD 001C3	26\$: PUSHL	R3	3156
	58	A8	9F 001C5	PUSHAB	P.AEK	
66		02	FB 001C8	CALLS	#2, CLISGET_VALUE	
2E		50	E9 001CB	BLBC	R0, 30\$	
1E		52	E9 001CE	BLBC	FIRST, 29\$	3158
		53	DD 001D1	PUSHL	R3	3160
67		01	FB 001D3	CALLS	#1, STRIP_NEGATOR	
DC	A4	50	D0 001D6	MOVL	R0, NODE_SWITCH	
		63	B5 001DA	TSTW	DESC	3161
		0D	13 001DC	BEQL	28\$	
		7E	D4 001DE	CLRL	-(SP)	3162
		53	DD 001E0	PUSHL	R3	
		0C	DD 001E2	PUSHL	#12	
65		03	FB 001E4	CALLS	#3, ADD_SYMBOL	
01		50	E8 001E7	27\$: BLBS	STATUS, -28\$	
			04 001EA	RET		
		52	D4 001EB	28\$: CLRL	FIRST	3163
		D4	11 001ED	BRB	26\$	3158
		7E	D4 001EF	29\$: CLRL	-(SP)	3165
		53	DD 001F1	PUSHL	R3	
		0C	DD 001F3	PUSHL	#12	
65		03	FB 001F5	CALLS	#3, ADD_SYMBOL	
C8		50	E8 001F8	BLBS	STATUS, -26\$	
			04 001FB	RET		
52		01	D0 001FC	30\$: MOVL	#1, FIRST	3175
		53	DD 001FF	31\$: PUSHL	R3	3176
	68	A8	9F 00201	PUSHAB	P.AEM	
66		02	FB 00204	CALLS	#2, CLISGET_VALUE	
31		50	E9 00207	BLBC	R0, 33\$	
0B		52	E9 0020A	BLBC	FIRST, 32\$	3178
		53	DD 0020D	PUSHL	R3	
67		01	FB 0020F	CALLS	#1, STRIP_NEGATOR	
E0	A4	50	D0 00212	MOVL	R0, OWNER_SWITCH	
		52	D4 00216	CLRL	FIRST	
50		63	3C 00218	32\$: MOVZWL	DESC, R0	3179
		E2	13 0021B	BEQL	31\$	
	04	A3	DD 0021D	PUSHL	DESC+4	3184
	04	A3	DD 00220	PUSHL	DESC+4	3183
		50	DD 00223	PUSHL	R0	3182
68		03	FB 00225	CALLS	#3, LIB\$CVT_HTB	
78		50	E9 00228	BLBC	R0, 38\$	
63		04	B0 0022B	MOVW	#4, DESC	3186
		7E	D4 0022E	CLRL	-(SP)	3187
		53	DD 00230	PUSHL	R3	
		0D	DD 00232	PUSHL	#13	
65		03	FB 00234	CALLS	#3, ADD_SYMBOL	
C5		50	E8 00237	BLBS	STATUS, -31\$	
			04 0023A	RET		
52		01	D0 0023B	33\$: MOVL	#1, FIRST	3196
		53	DD 0023E	34\$: PUSHL	R3	3197
	78	A8	9F 00240	PUSHAB	P.AEO	
66		02	FB 00243	CALLS	#2, CLISGET_VALUE	
31		50	E9 00246	BLBC	R0, 36\$	
0B		52	E9 00249	BLBC	FIRST, 35\$	3199
		53	DD 0024C	PUSHL	R3	

E4	67	01	FB	0024E	CALLS	#1, STRIP NEGATOR	
A4	50	50	DD	00251	MOVL	R0, PRIORITY_SWITCH	
	50	52	D4	00255	CLRL	FIRST	
		63	3C	00257	MOVZWL	DESC, R0	3200
		E2	13	0025A	BEQL	34\$	
		04	A3	DD	0025C	PUSHL	DESC+4
		04	A3	DD	0025F	PUSHL	DESC+4
			50	DD	00262	PUSHL	R0
	6A	03	FB	00264	CALLS	#3, LIB\$CVT_DTB	3205
	5A	50	E9	00267	BLBC	R0, 41\$	3204
	63	04	B0	0026A	MOVW	#4, DESC	3203
		7E	D4	0026D	CLRL	-(SP)	
		53	DD	0026F	PUSHL	R3	3207
		0F	DD	00271	PUSHL	#15	3208
	65	03	FB	00273	CALLS	#3, ADD_SYMBOL	
	C5	50	E8	00276	BLBS	STATUS, -34\$	
			04	00279	RET		
	52	01	DD	0027A	MOVL	#1, FIRST	3218
		53	DD	0027D	PUSHL	R3	3219
		C8	9F	0027F	PUSHAB	P.AEQ	
	66	02	FB	00283	CALLS	#2, CLISGET_VALUE	
	4F	50	E9	00286	BLBC	R0, 43\$	
	2E	52	E9	00289	BLBC	FIRST, 40\$	3223
		53	DD	0028C	PUSHL	R3	3224
E8	67	01	FB	0028E	CALLS	#1, STRIP NEGATOR	
A4	50	50	DD	00291	MOVL	R0, PROCESS_SWITCH	
	63	B5	00295	TSTW	DESC		3226
		1D	13	00297	BEQL	39\$	
		5E	DD	00299	PUSHL	SP	3228
		5C	A3	9F	0029B	PUSHAB	PROCESS_KEYWORD
			53	DD	0029E	PUSHL	R3
	69	03	FB	002A0	CALLS	#3, LIB\$LOOKUP_KEY	
	1E	50	E9	002A3	BLBC	STATUS, 41\$	
		7E	D4	002A6	CLRL	-(SP)	3229
		04	AE	DD	002A8	PUSHL	PROCESS_VALUE
			10	DD	002AB	PUSHL	#16
	65	03	FB	002AD	CALLS	#3, ADD_SYMBOL	
	03	50	E8	002B0	BLBS	STATUS, -39\$	
		0086	31	002B3	BRW	49\$	
		52	D4	002B6	CLRL	FIRST	3231
		C3	11	002B8	BRB	37\$	3232
		5E	DD	002BA	PUSHL	SP	3233
		5C	A3	9F	002BC	PUSHAB	PROCESS_KEYWORD
			53	DD	002BF	PUSHL	R3
	69	03	FB	002C1	CALLS	#3, LIB\$LOOKUP_KEY	
	03	50	E8	002C4	BLBS	STATUS, 42\$	
		01CA	31	002C7	BRW	72\$	
		7E	D4	002CA	CLRL	-(SP)	3235
		04	AE	DD	002CC	PUSHL	PROCESS_VALUE
			10	DD	002CF	PUSHL	#16
	65	03	FB	002D1	CALLS	#3, ADD_SYMBOL	
	A6	50	E8	002D4	BLBS	STATUS, -37\$	
			04	002D7	RET		
	52	01	DD	002D8	MOVL	#1, FIRST	3244
		53	DD	002DB	PUSHL	R3	3247
		0098	C8	9F	002DD	PUSHAB	P.AES
	66	02	FB	002E1	CALLS	#2, CLISGET_VALUE	

	2D		50	E9	002E4	BLBC	R0, 47\$		
	1D		52	E9	002E7	BLBC	FIRST, 46\$	3249	
			53	DD	002EA	PUSHL	R3	3251	
EC	67		01	FB	002EC	CALLS	#1, STRIP_NEGATOR		
	A4		50	D0	002EF	MOVL	R0, QUEUE_SWITCH		
			63	B5	002F3	TSTW	DESC	3252	
			0C	13	002F5	BEQL	45\$		
			7E	D4	002F7	CLRL	-(SP)	3253	
			53	DD	002F9	PUSHL	R3		
			11	DD	002FB	PUSHL	#17		
	65		03	FB	002FD	CALLS	#3, ADD_SYMBOL		
	52		50	E9	00300	BLBC	STATUS, -53\$		
			52	D4	00303	CLRL	FIRST	3254	
			D4	11	00305	BRB	44\$	3249	
			7E	D4	00307	CLRL	-(SP)	3256	
			53	DD	00309	PUSHL	R3		
			11	DD	0030B	PUSHL	#17		
	65		03	FB	0030D	CALLS	#3, ADD_SYMBOL		
	C8		50	E8	00310	BLBS	STATUS, -44\$		
				04	00313	RET			
	52		01	D0	00314	MOVL	#1, FIRST	3265	
			53	DD	00317	PUSHL	R3	3266	
		00AC	C8	9F	00319	PUSHAB	P.AEU		
	66		02	FB	0031D	CALLS	#2, CLISGET_VALUE		
	2D		50	E9	00320	BLBC	R0, 52\$		
	1D		52	E9	00323	BLBC	FIRST, 51\$	3268	
			53	DD	00326	PUSHL	R3	3270	
	67		01	FB	00328	CALLS	#1, STRIP_NEGATOR		
F0	A4		50	D0	0032B	MOVL	R0, REMOTE_ID_SWITCH		
			63	B5	0032F	TSTW	DESC	3271	
			0C	13	00331	BEQL	50\$		
			7E	D4	00333	CLRL	-(SP)	3272	
			53	DD	00335	PUSHL	R3		
			13	DD	00337	PUSHL	#19		
	65		03	FB	00339	CALLS	#3, ADD_SYMBOL		
	16		50	E9	0033C	BLBC	STATUS, -53\$		
			52	D4	0033F	CLRL	FIRST	3273	
			D4	11	00341	BRB	48\$	3268	
			7E	D4	00343	CLRL	-(SP)	3275	
			53	DD	00345	PUSHL	R3		
			13	DD	00347	PUSHL	#19		
	65		03	FB	00349	CALLS	#3, ADD_SYMBOL		
	C8		50	E6	0034C	BLBS	STATUS, -48\$		
				04	0034F	RET			
0000V	CF		00	FB	00350	CALLS	#0, PARSE_KEYS	3284	
	68		50	E9	00355	BLBC	STATUS, 58\$		
	52		01	D0	00358	MOVL	#1, FIRST	3291	
			53	DD	0035B	PUSHL	R3	3292	
		00BC	C8	9F	0035D	PUSHAB	P.AEW		
	66		02	FB	00361	CALLS	#2, CLISGET_VALUE		
	31		50	E9	00364	BLBC	R0, 56\$		
	0B		52	E9	00367	BLBC	FIRST, 55\$	3294	
			53	DD	0036A	PUSHL	R3		
	67		01	FB	0036C	CALLS	#1, STRIP_NEGATOR		
F4	A4		50	D0	0036F	MOVL	R0, STATUS_SWITCH		
			52	D4	00373	CLRL	FIRST		
	50		63	3C	00375	MOVZWL	DESC, R0	3295	

		E1	13	00378	BEQL	54\$		
	04	A3	DD	0037A	PUSHL	DESC+4		3301
	04	A3	DD	0037D	PUSHL	DESC+4		3300
		5C	DD	00380	PUSHL	R0		3299
68		03	FB	00382	CALLS	#3, LIB\$CVT_HTB		
76		50	E9	00385	BLBC	R0, 63\$		
63		04	B0	00388	MOVW	#4, DESC		3303
		7E	D4	0038B	CLRL	-(SP)		3304
		53	DD	0038D	PUSHL	R3		
		16	DD	0038F	PUSHL	#22		
65		03	FB	00391	CALLS	#3, ADD_SYMBOL		
C4		50	E8	00394	BLBS	STATUS_STATUS, 54\$		3305
			04	00397	RET			
52		01	DD	00398	56\$:	MOVL	#1, FIRST	3315
		53	DD	0039B	57\$:	PUSHL	R3	3316
	00CC	C8	9F	0039D	PUSHAB	P.AEY		
66		02	FB	003A1	CALLS	#2, CLISGET_VALUE		
2D		50	E9	003A4	BLBC	R0, 61\$		
1D		52	E9	003A7	BLBC	FIRST, 60\$		3318
		53	DD	003AA	PUSHL	R3		3320
67		01	FB	003AC	CALLS	#1, STRIP_NEGATOR		
F8	A4	50	DD	003AF	MOVL	R0, TERMINAL_SWITCH		
		63	B5	003B3	TSTW	DESC		3321
		0C	13	003B5	BEQL	59\$		
		7E	D4	003B7	CLRL	-(SP)		3322
		53	DD	003B9	PUSHL	R3		
		19	DD	003BB	PUSHL	#25		
65		03	FB	003BD	CALLS	#3, ADD_SYMBOL		
48		50	E9	003C0	58\$:	BLBC	STATUS, -64\$	
		52	D4	003C3	59\$:	CLRL	FIRST	3323
		D4	11	003C5	BRB	57\$		3318
		7E	D4	003C7	60\$:	CLRL	-(SP)	3325
		53	DD	003C9	PUSHL	R3		
		19	DD	003CB	PUSHL	#25		
65		03	FB	003CD	CALLS	#3, ADD_SYMBOL		
C8		50	E8	003D0	BLBS	STATUS, -57\$		
			04	003D3	RET			
52		01	DD	003D4	61\$:	MOVL	#1, FIRST	3334
		53	DD	003D7	62\$:	PUSHL	R3	3335
	00D8	C8	9F	003D9	PUSHAB	P.AFA		
66		02	FB	003DD	CALLS	#2, CLISGET_VALUE		
4C		50	E9	003E0	BLBC	R0, 68\$		
2D		52	E9	003E3	BLBC	FIRST, 66\$		3338
		53	DD	003E6	PUSHL	R3		3340
67		01	FB	003E8	CALLS	#1, STRIP_NEGATOR		
F8	A4	50	DD	003EB	MOVL	R0, TYPE_SWITCH		
		63	B5	003EF	TSTW	DESC		3341
		1C	13	003F1	BEQL	65\$		
	04	AE	9F	003F3	PUSHAB	TYPE_VALUE		3343
	08	A3	9F	003F6	PUSHAB	TYPE_KEYWORD		
		53	DD	003F9	PUSHL	R3		
69		03	FB	003FB	CALLS	#3, LIB\$LOOKUP_KEY		
1D		50	E9	003FE	63\$:	BLBC	STATUS, 67\$	
		7E	D4	00401	CLRL	-(SP)		3344
	08	AE	DD	00403	PUSHL	TYPE_VALUE		
		1B	DD	00406	PUSHL	#27		
65		03	FB	00408	CALLS	#3, ADD_SYMBOL		

00222D22	8F	04	B3	01	50	E8 0040B	64\$:	BLBS	STATUS, 65\$	:			
						04 0040E		RET		:			
					52	D4 0040F	65\$:	CLRL	FIRST	:	3346		
					C4	11 00411		BRB	62\$	:	3338		65
				04	AE	9F 00413	66\$:	PUSHAB	TYPE_VALUE	:	3349		2C
				09	A3	9F 00416		PUSHAB	TYPE_KEYWORD	:			
					53	DD 00419		PUSHL	R3	:			
				69	03	FB 0041B		CALLS	#3, LIB\$LOOKUP_KEY	:			
				73	50	E9 0041E	67\$:	BLBC	STATUS, 72\$	:			
					7E	D4 00421		CLRL	-(SP)	:	3350		
					08	AE DD 00423		PUSHL	TYPE_VALUE	:			
					1B	DD 00426		PUSHL	#27	:			
				65	03	FB 00428		CALLS	#3, ADD_SYMBOL	:			
				A9	50	E8 0042B		BLBS	STATUS, 62\$	:			
						04 0042E		RET		:			
				52	01	D0 0042F	68\$:	MOVL	#1, FIRST	:	3366		
				64	01	D0 00432		MOVL	#1, UIC_SWITCH	:	3367		
	04			A4	01	D0 00435		MOVL	#1, UIC_GROUP_SWITCH	:	3368		
	08			A4	01	D0 00439		MOVL	#1, UIC_MEMBER_SWITCH	:	3369		
					53	DD 0043D	69\$:	PUSHL	R3	:	3370		
					00E4	C8 9F 0043F		PUSHAB	P.AFC	:			
				66	02	FB 00443		CALLS	#2, CLISGET_VALUE	:			
				03	50	E8 00446		BLBS	RO, 70\$	:			
					00B4	31 00449		BRW	76\$	:			
				17	52	E9 0044C	70\$:	BLBC	FIRST, 71\$	:	3372		
					52	D4 0044F		CLRL	FIRST	:	3374		
				03	63	B1 00451		CMPL	DESC, #3	:	3375		
					10	12 00454		BNEQ	71\$	:			
				18	00	ED 00456		CMPL	#0, #24, @DESC+4, #2239778	:	3376		
					04	12 00460		BNEQ	71\$	:			
					64	D4 00462		CLRL	UIC_SWITCH	:	3378		2C
					63	B4 00464		CLRW	DESC	:	3379		2C
				50	63	3C 00466	71\$:	MOVZWL	DESC, RO	:	3382		2C
					D2	13 00469		BEQL	69\$	:			
	00A0			C3	50	D0 0046B		MOVL	RO, TPARSE_BLOCK+8	:	3384		
	00A4			C3	04	A3 D0 00470		MOVL	DESC+4, TPARSE_BLOCK+12	:	3385		
					FE48	C4 7C 00476		CLRW	GROUP	:	3386		
					00000000G	00 9F 0047A		PUSHAB	UIC_KTB	:	3388		
					00000000G	00 9F 00480		PUSHAB	UIC_STB	:			
					0098	C3 9F 00486		PUSHAB	TPARSE_BLOCK	:			
	00000000G			00	03	FB 0048A		CALLS	#3, LIB\$TPARSE	:			
				19	50	E8 00491		BLBS	RO, 73\$	:			0C
					53	DD 00494	72\$:	PUSHL	R3	:	3389		
					01	DD 00496		PUSHL	#1	:			
					009F10FC	8F DD 00498		PUSHL	#10424572	:			
	00000000G			00	03	FB 0049E		CALLS	#3, LIB\$SIGNAL	:			
				50	009F10FC	8F D0 004A5		MOVL	#10424572, RO	:			
						04 004AC		RET		:			
					FE48	C4 3C 004AD	73\$:	MOVZWL	CONVERTED_UIC+2, GROUP	:	3390		
					FE4C	C4 3C 004B4		MOVZWL	CONVERTED_UIC, MEMBER	:	3391		
	00003FFF			8F	FE48	C4 D1 004BB		CMPL	GROUP, #18383	:	3392		
						16 13 004C4		BEQL	74\$	:			
	00000000G			00	01	88 004C6		BISB2	#1, QUALIFIERS+4	:	3395		
				04	A4	D0 004CD		MOVL	UIC_SWITCH, UIC_GROUP_SWITCH	:	3396		
					7E	D4 004D1		CLRL	-(SP)	:	3397		
					0088	C3 9F 004D3		PUSHAB	GROUP_DESC	:			
					20	DD 004D7		PUSHL	#32	:			

ACC
V04-000

L 10
13-Sep-1984 23:36:11
14-Sep-1984 11:51:58

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[ACC.SRC]ACC.B32;1

Page 91
(14)

AC
V0

0000FFFF	65	FE4C	03	FB	004D9	CALLS	#3, ADD_SYMBOL	:	
	8F		C4	D1	004DC	CMPL	MEMBER, -#65535	:	3399
00000000G	00		16	13	004E5	BEQL	75\$	:	
08	A4		02	88	004E7	BISB2	#2, QUALIFIERS+4	:	3402
			64	D0	004EE	MOVL	UIC_SWITCH, UIC_MEMBER_SWITCH	:	3403
		0090	7E	D4	004F2	CLRL	-(SP)	:	3404
			C3	9F	004F4	PUSHAB	MEMBER_DESC	:	
	65		21	DD	004F8	PUSHL	#33	:	
			03	FB	004FA	CALLS	#3, ADD_SYMBOL	:	
		FF3D	31	004FD	75\$:	BRW	69\$	:	3370
	52		01	D0	00500	76\$:	MOVL	#1, FIRST	3414
			53	DD	00503	77\$:	PUSHL	R3	3415
		00F0	C8	9F	00505	PUSHAB	P.AFE	:	
	66		02	FB	00509	CALLS	#2, CLISGET_VALUE	:	
	2D		50	E9	0050C	BLBC	R0, 80\$	:	
	1D		52	E9	0050F	BLBC	FIRST, 79\$	:	3417
			53	DD	00512	PUSHL	R3	:	3419
	67		01	FB	00514	CALLS	#1, STRIP_NEGATOR	:	
0C	A4		50	D0	00517	MOVL	R0, USER_SWITCH	:	
			63	B5	0051B	TSTW	DESC	:	3420
			0C	13	0051D	BEQL	78\$	:	
			7E	D4	0051F	CLRL	-(SP)	:	3421
			53	DD	00521	PUSHL	R3	:	
			04	DD	00523	PUSHL	#4	:	
	65		03	FB	00525	CALLS	#3, ADD_SYMBOL	:	
	14		50	E9	00528	BLBC	STATUS, -81\$	:	
			52	D4	0052B	78\$:	CLRL	FIRST	3422
			D4	11	0052D	BRB	77\$	:	3417
			7E	D4	0052F	79\$:	CLRL	-(SP)	3424
			53	DD	00531	PUSHL	R3	:	
			04	DD	00533	PUSHL	#4	:	
	65		03	FB	00535	CALLS	#3, ADD_SYMBOL	:	
	C8		50	E8	00538	BLBS	STATUS, -77\$	:	
			04	0053B	RET			:	
	50		01	D0	0053C	80\$:	MOVL	#1, R0	3429
			04	0053F	81\$:	RET		:	3431

; Routine Size: 1344 bytes, Routine Base: CODE + 15D4

2C

2C

```
2608 3432 1 ROUTINE PARSE_DATES =
2609 3433 1
2610 3434 1 ----
2611 3435 1
2612 3436 1 Functional description
2613 3437 1
2614 3438 1 This routine is called to parse the /DATE, /BEFORE and /SINCE
2615 3439 1 command switches.
2616 3440 1
2617 3441 1 Input parameters
2618 3442 1
2619 3443 1 None
2620 3444 1
2621 3445 1 Output parameters
2622 3446 1
2623 3447 1
2624 3448 1 Store dates for future processing.
2625 3449 1 Set up an offset to the date in the record to be
2626 3450 1 used for displays and selection criteria.
2627 3451 1 Any errors encountered are RETURNed immediately.
2628 3452 1
2629 3453 1 ----
2630 3454 1
2631 3455 2 BEGIN
2632 3456 2
2633 3457 2 OWN
2634 3458 2 desc: bblock [dsc$k_d_bln] ! Dynamic work descriptor
2635 3459 2 preset([dsc$b_c[ass] = dsc$k_class_d),
2636 3460 2
2637 3461 2
2638 3462 2 DEFINE KEYWORD TABLES --
2639 3463 2 Define a keyword table for LIB$LOOKUP_KEY. Associate a byte
2640 3464 2 offset with each of the date keywords. This offset defines
2641 3465 2 the location within an accounting file of that date.
2642 3466 2
2643 3467 2
2644 P 3468 2 time_keyword: $lib_key_table ( ! Keyword table -- time type
2645 P 3469 2 (FINISHED, finish_time_desc [1]), ! -Time job terminated
2646 P 3470 2 (LOGOUT, finish_time_desc [1]), ! -Ditto
2647 P 3471 2 (STOPPED, finish_time_desc [1]), ! -Ditto
2648 P 3472 2
2649 P 3473 2 (STARTED, start_time_desc [1]), ! -Time job logged in
2650 P 3474 2 (LOGIN, start_time_desc [1]), ! -Ditto
2651 P 3475 2
2652 P 3476 2 (QUEUED, queue_time_desc [1]), ! -Time job was queued
2653 P 3477 2 (SUBMITTED, queue_time_desc [1]) ! -Ditto
2654 3478 2 );
2655 3479 2
2656 3480 2
2657 3481 2
2658 3482 2
2659 3483 2 PARSE /DATE VALUE --
2660 3484 2
2661 3485 2
2662 3486 2 If GET_VALUE ('DATE', desc) then ! /DATE type
2663 3487 2 LOOKUP (desc, time_keyword, date_desc_addr);
2664 3488 2
```

```
: 2665      3489 2 date_desc_addr = finish_time_desc [1];
: 2666      3490 2 date_header = addrdesc ('finished');          ! Set default date header
: 2667      3491 2
: 2668      3492 2 ! If .date_desc_addr eql queue_time_desc [1]      ! If /DATE=QUEUED
: 2669      3493 2 !   then date_header = addrdesc ('queued');      ! Fix date header
: 2670      3494 2
: 2671      3495 2 ! If .date_desc_addr eql start_time_desc [1]      ! If /DATE=STARTED
: 2672      3496 2 !   then date_header = addrdesc ('started');      ! Fix date header
: 2673      3497 2
: 2674      3498 2
: 2675      3499 2 ! PARSE /BEFORE AND /SINCE DATE VALUES --
: 2676      3500 2
: 2677      3501 2
: 2678      3502 2 If GET_VALUE ('BEFORE', desc) then          ! /BEFORE value
: 2679      3503 2     perform (lib$cvtime (desc, before_date));
: 2680      3504 2
: 2681      3505 2 If GET_VALUE ('SINCE', desc) then          ! /SINCE value
: 2682      3506 2     perform (lib$cvtime (desc, since_date));
: 2683      3507 2
: 2684      3508 2 Return true;
: 2685      3509 1 END;
```

```
00 00 00 44 45 48 53 49 4E 49 46 08 01B14 P.AFG: .ASCII <8>\FINISHED\<0><0><0>
00 54 55 4F 47 4F 4C 06 01B20 P.AFH: .ASCII <6>\LOGOUT\<0>
44 45 50 50 4F 54 53 07 01B28 P.AFI: .ASCII <7>\STOPPED\
44 45 54 52 41 54 53 07 01B30 P.AFJ: .ASCII <7>\STARTED\
00 00 4E 49 47 4F 4C 05 01B38 P.AFK: .ASCII <5>\LOGIN\<0><0>
00 44 45 55 45 55 51 06 01B40 P.AFL: .ASCII <6>\QUEUED\<0>
00 00 44 45 54 54 49 4D 42 55 53 09 01B48 P.AFM: .ASCII <9>\SUBMITTED\<0><0>
00 00 44 45 54 54 49 4D 42 55 53 09 01B54 P.AFO: .ASCII \finished\
64 65 68 73 69 6E 69 66 01B5C P.AFN: .LONG 8
00000008 01B60 .ADDRESS P.AFO
00 00 45 52 4F 46 45 42 01B64 P.AFQ: .ASCII \BEFORE\<0><0>
00000006 01B6C P.AFP: .LONG 6
00000006 01B70 .ADDRESS P.AFQ
00 00 00 45 43 4E 49 53 01B74 P.AFS: .ASCII \SINCE\<0><0><0>
00000005 01B7C P.AFR: .LONG 5
00000000 01B80 .ADDRESS P.AFS
```

.PSECT \$OWNS,NOEXE,2

```
00# 000CC DESC: .BYTE 0[3]
02 000CF .BYTE 2
000D0 .BLKB 4
0000000E 000D4 TIME_KEYWORD:
.LONG 14
.ADDRESS P.AFG, FINISH_TIME_DESC+4, P.AFH, -
FINISH_TIME_DESC+4, P.AFI, -
FINISH_TIME_DESC+4, P.AFJ, -
START_TIME_DESC+4, P.AFK, -
START_TIME_DESC+4, P.AFL, -
QUEUE_TIME_DESC+4, P.AFM, -
QUEUE_TIME_DESC+4
```

00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 000D8
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 000F0
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00108

.PSECT CODE, NOWRT, 2

```
003C 00000 PARSE_DATES:
55 00000000G 00 9E 00002 .WORD Save R2,R3,R4,R5 ; 3432
54 00000000G 00 9E 00009 MOVAB LIB$CVT_TIME, R5 ;
53 0000' CF 9E 00010 MOVAB CL$GET_VALUE, R4 ;
52 00000000' EF 9E 00015 MOVAB DESC, R3 ;
0398 C2 62 9E 0001C MOVAB FINISH_TIME_DESC+4, R2 ;
0168 C2 B4 AF 9E 00021 MOVAB FINISH_TIME_DESC+4, DATE_DESC_ADDR ; 3489
BC 53 DD 00027 MOVAB P.AFN, DATE_HEADER ; 3490
AF 9F 00029 PUSHAB P.AFP ; 3502
64 02 FB 0002C CALLS #2, CL$GET_VALUE ;
0C 50 E9 0002F BLBC R0, 1$ ;
039C C2 9F 00032 PUSHAB BEFORE_DATE ; 3503
53 DD 00036 PUSHAB R3 ;
65 02 FB 00038 CALLS #2, LIB$CVT_TIME ;
1A 50 E9 0003B BLBC STATUS, 3$ ;
53 DD 0003E 1$: PUSHAB R3 ; 3505
B5 AF 9F 00040 PUSHAB P.AFR ;
64 02 FB 00043 CALLS #2, CL$GET_VALUE ;
0C 50 E9 00046 BLBC R0, 2$ ;
03A4 C2 9F 00049 PUSHAB SINCE_DATE ; 3506
53 DD 0004D PUSHAB R3 ;
65 02 FB 0004F CALLS #2, LIB$CVT_TIME ;
03 50 E9 00052 BLBC STATUS, 3$ ;
50 01 DD 00055 2$: MOVL #1, R0 ; 3508
04 00058 3$: RET ; 3509
```

; Routine Size: 89 bytes, Routine Base: CODE + 1B84


```

: 2687 3510 1 ROUTINE PARSE_KEYS =
: 2688 3511 1
: 2689 3512 1 ---
: 2690 3513 1
: 2691 3514 1 Functional description
: 2692 3515 1
: 2693 3516 1 This routine is called from the main parser to handle those
: 2694 3517 1 qualifiers that causes dynamic strings to be built. That is,
: 2695 3518 1 sort keys, report items, and summarization keys.
: 2696 3519 1
: 2697 3520 1 Output parameters
: 2698 3521 1
: 2699 3522 1 Any errors encountered are RETURNed immediately.
: 2700 3523 1 TRUE is returned on a normal exit.
: 2701 3524 1
: 2702 3525 1 ----
: 2703 3526 1
: 2704 3527 2 BEGIN
: 2705 3528 2
: 2706 3529 2 LOCAL
: 2707 3530 2 result: ref vector [2]; ! Holds pointer returned by lookup
: 2708 3531 2
: 2709 3532 2
: 2710 3533 2 Each of the three element entries below consists of the following:
: 2711 3534 2
: 2712 3535 2 1) Key type (1=char, 2=binary)
: 2713 3536 2 3) Address of record item descriptor
: 2714 3537 2 4) Maximum size in bytes of the target field
: 2715 3538 2
: 2716 3539 2 GLOBAL
: 2717 3540 2
: 2718 3541 2 key_table: blockvector [32, 3] initial (
: 2719 3542 2
: 2720 3543 2 1. account_desc, 8, ! Account name
: 2721 3544 2 2. entry_desc, 4, ! Entry (job id)
: 2722 3545 2 2. finish_time_desc, 8, ! Termination time
: 2723 3546 2 2. ident_desc, 4, ! Process ID
: 2724 3547 2 1. image_desc, 9, ! Image name
: 2725 3548 2 1. job_desc, 39, ! Job name
: 2726 3549 2 1. node_desc, 6, ! Remote node name
: 2727 3550 2 2. owner_desc, 4, ! Owner process ID
: 2728 3551 2 1. process_desc, 11, ! Process type
: 2729 3552 2 1. queue_desc, 31, ! Queue name
: 2730 3553 2 2. queue_time_desc, 8, ! Queue time
: 2731 3554 2 2. start_time_desc, 8, ! Start time
: 2732 3555 2 2. status_desc, 4, ! Final status
: 2733 3556 2 1. terminal_desc, 8, ! Terminal name
: 2734 3557 2 1. type_desc, 8, ! Record type
: 2735 3558 2 1. user_desc, 12, ! User name
: 2736 3559 2 2. address_desc, 4, ! Remote node address
: 2737 3560 2 2. buffered_io_desc, 4, ! Buffered IO count
: 2738 3561 2 2. direct_io_desc, 4, ! Direct IO count
: 2739 3562 2 2. cpu_time_desc, 4, ! Cpu time
: 2740 3563 2 2. page_faults_desc, 4, ! Page faults
: 2741 3564 2 2. page_reads_desc, 4, ! Page IO
: 2742 3565 2 2. page_file_desc, 4, ! Page file peak
: 2743 3566 2 2. working_set_desc, 4, ! Working set peak
```

```
2744 3567 2 2. execution_desc, 4,      Image execution count
2745 3568 2 2. elapsed_desc, 8,      Elapsed time
2746 3569 2 2. priority_desc, 1,      Priority
2747 3570 2 2. volumes_desc, 4,      Volumes mounted
2748 3571 2 2. sym_page_desc, 4,      Pages printed
2749 3572 2 2. sym_get_desc, 4,      Symbiont gets
2750 3573 2 2. sym_qio_desc, 4,      Symbiont qios
2751 3574 2 2. uic_binary_desc, 4,    UIC
2752 3575 2
2753 3576 2
2754 3577 2 OWN
2755 3578 2 desc: bblock [dsc$k_d_bln] ! Dynamic work descriptor
2756 3579 2 preset([dsc$b_class] = dsc$k_class_d),
2757 3580 2
2758 3581 2
2759 3582 2 ! Now build a table that associates each sort keyword
2760 3583 2 ! with one of the sort key descriptor elements
2761 3584 2 !
2762 3585 2
2763 P 3586 2 sort_keyword: $lib_key_table ( ! Keywords for /SORT qualifier
2764 P 3587 2
2765 P 3588 2 (ACCOUNT, 00), ! User's account name
2766 P 3589 2 (ENTRY, 01), ! Number of batch or print job
2767 P 3590 2 (FINISHED, 02), ! Termination time
2768 P 3591 2 (IDENT, 03), ! Process identification
2769 P 3592 2 (IMAGE, 04), ! Image name
2770 P 3593 2 (JOB, 05), ! Name of batch or print job
2771 P 3594 2 (NODE, 06), ! Node name
2772 P 3595 2 (OWNER, 07), ! Owner process identification
2773 P 3596 2 (PROCESS, 08), ! Process type
2774 P 3597 2 (QUEUE, 09), ! Name of queue
2775 P 3598 2 (QUEUED, 10), ! Time job was queued
2776 P 3599 2 (STARTED, 11), ! Start time
2777 P 3600 2 (STATUS, 12), ! Final exit status
2778 P 3601 2 (TERMINAL, 13), ! Terminal name
2779 P 3602 2 (TYPE, 14), ! Record type
2780 P 3603 2 (USER, 15), ! User's name
2781 P 3604 2 (ADDRESS, 16), ! Remote node address
2782 P 3605 2 (BUFFERED IO, 17), ! Buffered IO count
2783 P 3606 2 (DIRECT IO, 18), ! Direct IO count
2784 P 3607 2 (PROCESSOR, 19), ! CPU time
2785 P 3608 2 (FAULTS, 20), ! Page faults
2786 P 3609 2 (PAGE_READS, 21), ! Page IO
2787 P 3610 2 (PAGE_FILE, 22), ! Page file peak
2788 P 3611 2 (WORKING SET, 23), ! Working set peak
2789 P 3612 2 (EXECUTION, 24), ! Image execution count
2790 P 3613 2 (ELAPSED, 25), ! Elapsed time
2791 P 3614 2 (PRIORITY, 26), ! Priority
2792 P 3615 2 (VOLUMES, 27), ! Volumes mounted
2793 P 3616 2 (PAGES, 28), ! Pages printed
2794 P 3617 2 (GETS, 29), ! Symbiont gets
2795 P 3618 2 (QIOS, 30), ! Symbiont qios
2796 P 3619 2 (UIC, 31), ! UIC
2797 3620 2 )
2798 3621 2
2799 3622 2
2800 3623 2 !
```

```
2801 3624 2 ! Build a lookup table that associates each summary option keyword with
2802 3625 2 ! a three longword structure. The first longword contains the address
2803 3626 2 ! of a descriptor of the data field in the record. The second longword
2804 3627 2 ! contains the address of a descriptor of a FAO control string that
2805 3628 2 ! is used to concatenate keys. The third longword contains the address
2806 3629 2 ! of a descriptor of a string suitable for use in the header FAO.
2807 3630 2
2808 3631 2
2809 3632 2
2810 P 3633 2 summary_keyword: $lib_key_table ( ! Keywords for /SUMMARY qualifier
2811 P 3634 2
2812 P 3635 2 (ACCOUNT, UPLIT (account_desc, ad('!9AS'), ad('Account '))), ! Account
2813 P 3636 2 (DATE, UPLIT (s_date_desc, ad('!11AS'), ad('YYYY MM DD '))), ! Date
2814 P 3637 2 (DAY, UPLIT (day_desc, ad('!3AS'), ad('DD '))), ! Day of month
2815 P 3638 2 (HOUR, UPLIT (hour_desc, ad('!3AS'), ad('HH '))), ! Hour of day
2816 P 3639 2 (IMAGE, UPLIT (image_desc, ad('!11AS'), ad('Image name '))), ! Image name
2817 P 3640 2 (JOB, UPLIT (job_desc, ad('!39AS'), ad('Job name '))), ! Batch/print
2818 P 3641 2 (MONTH, UPLIT (month_desc, ad('!3AS'), ad('MM '))), ! Month of year
2819 P 3642 2 (NODE, UPLIT (node_desc, ad('!10AS'), ad('Node name '))), ! Node name
2820 P 3643 2 (PROCESS, UPLIT (process_desc, ad('!13AS'), ad('Process type '))), ! Process
2821 P 3644 2 (QUEUE, UPLIT (queue_desc, ad('!31AS'), ad('Job/print queue '))), ! Queue
2822 P 3645 2 (TERMINAL, UPLIT (terminal_desc, ad('!9AS'), ad('Terminal '))), ! Terminal name
2823 P 3646 2 (TYPE, UPLIT (type_desc, ad('!8AS'), ad('Type '))), ! Record type
2824 P 3647 2 (UIC, UPLIT (uic_desc, ad('!67AS'), ad('U I C '))), ! User
2825 P 3648 2 (USER, UPLIT (user_desc, ad('!13AS'), ad('Username '))), ! User
2826 P 3649 2 (WEEKDAY, UPLIT (weekday_desc, ad('!3AS'), ad('DD '))), ! Day of week
2827 P 3650 2 (YEAR, UPLIT (year_desc, ad('!5AS'), ad('Year '))), ! Year
2828 3651 2
2829 3652 2
2830 3653 2
2831 3654 2 ! Build a table that associates each report option keyword with
2832 3655 2 ! a structure of longwords that contains:
2833 3656 2
2834 3657 2 1) Summation type (additive, peak, elapsed time, etc)
2835 3658 2 2) Summation value (address of value, constant, etc)
2836 3659 2 3) Number of longwords needed in accumulator
2837 3660 2 4) Descriptor of FAO control string for eventual output
2838 3661 2 5) Descriptor of FAO control string for title line 1
2839 3662 2 6) Descriptor of FAO control string for title line 2
2840 3663 2
2841 3664 2
2842 P 3665 2 report_keyword: $lib_key_table ( ! Keywords for /REPORT qualifier
2843 P 3666 2
2844 P 3667 2 (BUFFERED_IO, UPLIT (sum_type_add, hold_biocnt, 1,
2845 P 3668 2 ad('!8UL'), ad('Buffered'), ad(' I/O '))),
2846 P 3669 2
2847 P 3670 2 (DIRECT_IO, UPLIT (sum_type_add, hold_diocnt, 1,
2848 P 3671 2 ad('!8UL'), ad('Direct'), ad(' I/O '))),
2849 P 3672 2
2850 P 3673 2 (ELAPSED, UPLIT (sum_type_addx, hold_elapsed, 4,
2851 P 3674 2 ad('!4UL-!1T'), ad('Elapsed ')),
2852 P 3675 2 ad(' Time '))),
2853 P 3676 2
2854 P 3677 2 (EXECUTION, UPLIT (sum_type_add, hold_execution, 1,
2855 P 3678 2 ad('!8UL'), ad('Image '), ad(' Count '))),
2856 P 3679 2
2857 P 3680 2 (FAULTS, UPLIT (sum_type_add, hold_pageflts, 1,
```

```
2858 P 3681 2 ad (' .8UL'), ad (' Page '), ad (' Faults'))),
2859 P 3682 2
2860 P 3683 2 (GETS, UPLIT (sum_type_add, hold_getcnt, 1,
2861 P 3684 2 ad (' .8UL'), ad (' Symbiont'), ad (' Gets '))),
2862 P 3685 2
2863 P 3686 2 (PAGES, UPLIT (sum_type_add, hold_pagcnt, 1,
2864 P 3687 2 ad (' .8UL'), ad (' Pages'), ad (' Printed'))),
2865 P 3688 2
2866 P 3689 2 (PAGE_FILE, UPLIT (sum_type_peak, hold_pgflpeak, 1,
2867 P 3690 2 ad (' .8UL'), ad (' Peakpage'), ad (' File use'))),
2868 P 3691 2
2869 P 3692 2 (PAGE_READS, UPLIT (sum_type_add, hold_pagereads, 1,
2870 P 3693 2 ad (' .8UL'), ad (' Page IO '), ad (' Reads '))),
2871 P 3694 2
2872 P 3695 2 (PROCESSOR, UPLIT (sum_type_addx, hold_cputim, 4,
2873 P 3696 2 ad (' .4UL'), ad (' Processor '),
2874 P 3697 2 ad (' Time '))),
2875 P 3698 2
2876 P 3699 2 (QIOS, UPLIT (sum_type_add, hold_qiocnt, 1,
2877 P 3700 2 ad (' .8UL'), ad (' Symbiont'), ad (' QIO's '))),
2878 P 3701 2
2879 P 3702 2 (RECORDS, UPLIT (sum_type_incr, 0, 1,
2880 P 3703 2 ad (' .8UL'), ad (' Total'), ad (' Records'))),
2881 P 3704 2
2882 P 3705 2 (VOLUMES, UPLIT (sum_type_add, hold_volumes, 1,
2883 P 3706 2 ad (' .8UL'), ad (' Volumes'), ad (' Mounted'))),
2884 P 3707 2
2885 P 3708 2 (WORKING_SET, UPLIT (sum_type_peak, hold_wspeak, 1,
2886 P 3709 2 ad (' .8UL'), ad (' Peakwork'), ad (' Set page'))
2887 P 3710 2
2888 3711 2 );
2889 3712 2
2890 3713 2
2891 3714 2
2892 3715 2
2893 3716 2 ! PARSE /SORT VALUE LIST --
2894 3717 2 Fetch the value(s) associated with the /SORT qualifier. Use
2895 3718 2 LIB$LOOKUP_KEY to parse them. If successful, an index into
2896 3719 2 a sort key descriptor table is returned. The descriptors are
2897 3720 2 copied (in the user specified order) to the SORT_TABLE.
2898 3721 2
2899 3722 2 Note that the first entry in SORT_TABLE is NOT used to store
2900 3723 2 the number and type of entries as is done with the other tables.
2901 3724 2 The sort keys must be preceded by a single word (SORT_KEY_COUNT)
2902 3725 2 that gives the number of keys.
2903 3726 2
2904 3727 2 sort_key_count = 0; ! Initialize sort key count
2905 3728 2 Incr i from 0 to max_sort do
2906 3729 2 BEGIN ! /SORT value list
2907 3730 2 Local key_order, index;
2908 3731 2
2909 3732 2 If not (GET_VALUE ('SORT', desc))
2910 3733 2 then exitloop;
2911 3734 2
2912 3735 2 Key_order = 0; ! Assume ascending
2913 3736 2 If not strip_negator (desc) then
2914 3737 2 key_order = 1; ! No, set descending
```

```
2915 3738 3
2916 3739 3 LOOKUP (desc, sort_keyword, index);
2917 3740 3
2918 3741 3
2919 3742 3
2920 3743 3
2921 3744 3 MOVE KEY DESCRIPTORS
2922 3745 3 Move the key descriptors. Note that more than one descriptor
2923 3746 3 may be needed to describe a key. The number of descriptors
2924 3747 3 is stored in the order field (ascend/descend) of the first
2925 3748 3 descriptor since this field is dynamically set based on the negator.
2926 3749 3
2927 3750 3
2928 3751 3 Sort_table [.sort_key_count, key_w_type] = .key_table [.index, sort_type];
2929 3752 3 Sort_table [.sort_key_count, key_w_order] = .key_order;
2930 3753 3 Sort_table [.sort_key_count, key_w_pos] = .sort_key_size + 1;
2931 3754 3 Sort_table [.sort_key_count, key_w_length] = .key_table [.index, sort_length];
2932 3755 3 Sort_index_table [.sort_key_count] = .index;
2933 3756 3 Sort_key_size = .sort_key_size + .key_table [.index, sort_length];
2934 3757 3 Sort_key_count = .sort_key_count + 1;
2935 3758 3
2936 3759 3 END;
2937 3760 3
2938 3761 3
2939 3762 3
2940 3763 3
2941 3764 3
2942 3765 3 PARSE /SUMMARY VALUE LIST --
2943 3766 3 Parse the list of values associated with the /SUMMARY qualifier.
2944 3767 3 For each value store the address of the value descriptor in a
2945 3768 3 table and build a concatenated FAO control string.
2946 3769 3
2947 3770 3 In build_summary the FAO control string and the list of descriptors
2948 3771 3 is used to build the summation key.
2949 3772 3
2950 3773 3 Also, the FAO control string which prints the column header is
2951 3774 3 modified to reflect the contents of the summation key.
2952 3775 3
2953 3776 3
2954 3777 3 Incra i do
2955 3778 3   if not GET_VALUE ('SUMMARY', desc) then exitloop ! Get value
2956 3779 3   else BEGIN
2957 3780 3     If str$compare (desc, ad ('USER')) eqlu 0 ! Special case
2958 3781 3     AND not PRESENT (TYPE)
2959 3782 3     then BEGIN
2960 3783 3       EXTERNAL qualifiers: bitvector []; ! Establish def type
2961 3784 3       qualifiers [TYPE] = true;
2962 3785 3       perform (add_symbol (type, ad ('LOGFAIL'), true));
2963 3786 3     END;
2964 3787 3
2965 3788 3 If str$compare (desc, ad ('UIC')) eqlu 0 ! If /SUMMARY=UIC
2966 3789 3 then ! then set a flag.
2967 3790 3   SUMMARY_UIC = 1;
2968 3791 3
2969 3792 3 If str$compare (desc, ad ('HOURL')) eqlu 0
2970 3793 3 or str$compare (desc, ad ('WEEKDAY')) eqlu 0
2971 3794 3 or str$compare (desc, ad ('DAY')) eqlu 0
```

```
: 2972      3795 3 or str$compare (desc, ad('MONTH')) eqlu 0
: 2973      3796 3 or str$compare (desc, ad('YEAR')) eqlu 0
: 2974      3797 3 or str$compare (desc, ad('DATE')) eqlu 0
: 2975      3798 4 then BEGIN
: 2976      3799 4   EXTERNAL qualifiers: bitvector [];
: 2977      3800 4   qualifiers [expand_date] = TRUE;
: 2978      3801 3   END;
: 2979      3802 3
: 2980      3803 3   LOOKUP (desc, summary_keyword, result);
: 2981      3804 3   sum_key_value [.i] = .result [0];
: 2982      3805 3   perform (str$append (sum_key_fao, .result [1]));
: 2983      3806 3   perform (str$append (report_hdr1_fao, .result [2]));
: 2984      3807 3   perform (str$append (report_hdr3_fao, .result [2]));
: 2985      3808 3   END;
: 2986      3809 2
: 2987      3810 2   ! Fill the front of report header 2 with spaces equal to the length
: 2988      3811 2   ! of the symbol header (which is equal to length of report header 1)
: 2989      3812 2
: 2990      3813 2
: 2991      3814 2 P Perform (str$dupl_char (
: 2992      3815 2   report_hdr2_fao, %REF (.report_hdr1_fao [dsc$w_length]));
: 2993      3816 2
: 2994      3817 2
: 2995      3818 2
: 2996      3819 2   PARSE /REPORT VALUE LIST --
: 2997      3820 2   Parse the list of values associated with the /REPORT qualifier.
: 2998      3821 2   For each value store the address of the control structure in the
: 2999      3822 2   value table and build concatenated FAO control strings for
: 3000      3823 2   report writing time. One string describes the header for the
: 3001      3824 2   report, the other the detail line.
: 3002      3825 2
: 3003      3826 2
: 3004      3827 2   ! Place a FAO argument at the front of the report detail FAO control
: 3005      3828 2   ! string to print the summation key at WRITE_SUMMARY time.
: 3006      3829 2
: 3007      3830 2 Perform (str$append (report_det_fao, ad('!AS')));
: 3008      3831 2
: 3009      3832 2 Incra i do
: 3010      3833 2   if not GET_VALUE ('REPORT', desc) then exitloop
: 3011      3834 2   else BEGIN
: 3012      3835 3   LOOKUP (desc, report_keyword, result);
: 3013      3836 3   report_value [.i] = .result;
: 3014      3837 3   bucket_size = .bucket_size + (.result [sum_ent_bsize] * 4);
: 3015      3838 3   perform (str$append (report_det_fao, .result [sum_ent_fao]));
: 3016      3839 3   perform (str$append (report_hdr1_fao, .result [sum_ent_hdr1]));
: 3017      3840 3   perform (str$append (report_hdr2_fao, .result [sum_ent_hdr2]));
: 3018      3841 3   report_items = .report_items + 1;
: 3019      3842 3   END;
: 3020      3843 2
: 3021      3844 2
: 3022      3845 2   ! If /REPORT is active then reserve one more slot in the bucket
: 3023      3846 2   ! for the symbol descriptor address
: 3024      3847 2
: 3025      3848 2
: 3026      3849 2 If .bucket_size neq 0 then bucket_size = .bucket_size + 4;
: 3027      3850 2
: 3028      3851 2 return true;
```

```
00 00 00 44 54 4E 55 4F 43 43 41 07 01BDD .BLKB 3
00 00 00 44 00 00 59 52 54 4E 45 05 01BE0 P.AFT: .ASCII <7>\ACCOUNT\
00 00 00 44 45 48 53 49 4E 49 46 08 01BE8 P.AFU: .ASCII <5>\ENTRY\<0><0>
00 00 00 44 00 00 54 4E 45 44 49 05 01BF0 P.AFV: .ASCII <8>\FINISHED\<0><0><0>
00 00 00 44 00 00 45 47 41 4D 49 05 01BFC P.AFW: .ASCII <5>\IDENT\<0><0>
00 00 00 44 00 00 45 47 41 4D 49 05 01C04 P.AFX: .ASCII <5>\IMAGE\<0><0>
00 00 00 44 00 00 45 47 41 4D 49 05 01C0C P.AFY: .ASCII <3>\JOB\
00 00 00 44 00 00 45 47 41 4D 49 05 01C10 P.AFZ: .ASCII <4>\NODE\<0><0><0>
00 00 00 44 00 00 52 45 4E 57 4F 05 01C18 P.AGA: .ASCII <5>\OWNER\<0><0>
00 00 00 44 53 53 45 43 4F 52 50 07 01C20 P.AGB: .ASCII <7>\PROCESS\
00 00 00 44 00 00 45 55 45 55 51 05 01C28 P.AGC: .ASCII <5>\QUEUE\<0><0>
00 00 00 44 00 00 45 55 45 55 51 06 01C30 P.AGD: .ASCII <6>\QUEUED\<0>
00 00 00 44 00 00 45 55 45 55 51 06 01C38 P.AGE: .ASCII <7>\STARTED\
00 00 00 44 00 00 45 55 45 55 51 06 01C40 P.AGF: .ASCII <6>\STATUS\<0>
00 00 00 44 00 00 45 55 45 55 51 06 01C48 P.AGG: .ASCII <8>\TERMINAL\<0><0><0>
00 00 00 44 00 00 45 55 45 55 51 06 01C54 P.AGH: .ASCII <4>\TYPE\<0><0><0>
00 00 00 44 00 00 45 55 45 55 51 06 01C5C P.AGI: .ASCII <4>\USER\<0><0><0>
00 00 00 44 00 00 45 55 45 55 51 06 01C64 P.AGJ: .ASCII <7>\ADDRESS\
00 00 00 44 00 00 45 55 45 55 51 06 01C6C P.AGK: .ASCII <11>\BUFFERED IO\
00 00 00 44 00 00 45 55 45 55 51 06 01C78 P.AGL: .ASCII <9>\DIRECT IO\<0><0>
00 00 00 44 00 00 45 55 45 55 51 06 01C84 P.AGM: .ASCII <9>\PROCESSOR\<0><0>
00 00 00 44 00 00 45 55 45 55 51 06 01C90 P.AGN: .ASCII <6>\FAULTS\<0>
00 00 00 44 00 00 45 55 45 55 51 06 01C98 P.AGO: .ASCII <10>\PAGE READS\<0>
00 00 00 44 00 00 45 55 45 55 51 06 01CA4 P.AGP: .ASCII <9>\PAGE FILE\<0><0>
00 00 00 44 00 00 45 55 45 55 51 06 01CB0 P.AGQ: .ASCII <11>\WORKING SET\
00 00 00 44 00 00 45 55 45 55 51 06 01CBC P.AGR: .ASCII <9>\EXECUTION\<0><0>
00 00 00 44 00 00 45 55 45 55 51 06 01CC8 P.AGS: .ASCII <7>\ELAPSED\
00 00 00 44 00 00 45 55 45 55 51 06 01CD0 P.AGT: .ASCII <8>\PRIORITY\<0><0><0>
00 00 00 44 00 00 45 55 45 55 51 06 01CDC P.AGU: .ASCII <7>\VOLUMES\
00 00 00 44 00 00 45 55 45 55 51 06 01CE4 P.AGV: .ASCII <5>\PAGES\<0><0>
00 00 00 44 00 00 45 55 45 55 51 06 01CEC P.AGW: .ASCII <4>\GETS\<0><0><0>
00 00 00 44 00 00 45 55 45 55 51 06 01CF4 P.AGX: .ASCII <4>\QIOS\<0><0><0>
00 00 00 44 00 00 45 55 45 55 51 06 01CFC P.AGY: .ASCII <3>\UIC\
00 00 00 44 00 00 45 55 45 55 51 06 01D00 P.AGZ: .ASCII <7>\ACCOUNT\
00 00 00 44 00 00 45 55 45 55 51 06 01D08 P.AHC: .ASCII \!9AS\
00 00 00 44 00 00 45 55 45 55 51 06 01D0C P.AHB: .LONG 4
00 00 00 44 00 00 45 55 45 55 51 06 01D10 P.AHE: .ADDRESS P.AHC
00 00 00 44 00 00 45 55 45 55 51 06 01D14 P.AHE: .ASCII \Account \<0><0><0>
00 00 00 44 00 00 45 55 45 55 51 06 01D20 P.AHD: .LONG 9
00 00 00 44 00 00 45 55 45 55 51 06 01D24 P.AHA: .ADDRESS P.AHE
00 00 00 44 00 00 45 55 45 55 51 06 01D28 P.AHA: .ADDRESS ACCOUNT DESC, P.AHB, P.AHD
00 00 00 44 00 00 45 55 45 55 51 06 01D34 P.AHF: .ASCII <4>\DATE\<0><0><0>
00 00 00 44 00 00 45 55 45 55 51 06 01D3C P.AHI: .ASCII \!11AS\<0><0><0>
00 00 00 44 00 00 45 55 45 55 51 06 01D44 P.AHH: .LONG 5
00 00 00 44 00 00 45 55 45 55 51 06 01D48 P.AHI: .ADDRESS P.AHI
00 00 00 44 00 00 45 55 45 55 51 06 01D4C P.AHK: .ASCII \YYYY MM DD \<0>
00 00 00 44 00 00 45 55 45 55 51 06 01D58 P.AHJ: .LONG 11
00 00 00 44 00 00 45 55 45 55 51 06 01D5C P.AHK: .ADDRESS P.AHK
00 00 00 44 00 00 45 55 45 55 51 06 01D60 P.AHG: .ADDRESS $ DATE DESC, P.AHH, P.AHJ
00 00 00 44 00 00 45 55 45 55 51 06 01D6C P.AHL: .ASCII <3>\DAY\
00 00 00 44 00 00 45 55 45 55 51 06 01D70 P.AHO: .ASCII \!3AS\
00 00 00 44 00 00 45 55 45 55 51 06 01D74 P.AHN: .LONG 4
```

```
00 20 00000000' 01D78 .ADDRESS P.AHO
00 20 44 44 01D7C P.AHQ: .ASCII \DD \<0>
00000003' 01D80 P.AHP: .LONG 3
00000000' 01D84 .ADDRESS P.AHO
00000000' 01D88 P.AHM: .ADDRESS DAY DESC, P.AHN, P.AHP
00 00 00 52 55 4F 48 04 01D94 P.AHR: .ASCII <4>\HOUR\<0><0><0>
53 41 33 21 01D9C P.AHU: .ASCII \!3AS\
00000004' 01DA0 P.AHT: .LONG 4
00000000' 01DA4 .ADDRESS P.AHU
00 20 48 48 01DA8 P.AHW: .ASCII \HH \<0>
00000003' 01DAC P.AHV: .LONG 3
00000000' 01DB0 .ADDRESS P.AHW
00000000' 01DB4 P.AHS: .ADDRESS HOUR DESC, P.AHT, P.AHV
00 00 45 47 41 4D 49 05 01DC0 P.AHX: .ASCII <5>\IMAGE\<0><0>
00 00 00 53 41 31 31 21 01DC8 P.AIA: .ASCII \!11AS\<0><0><0>
00000005' 01DD0 P.AHZ: .LONG 5
00000000' 01DD4 .ADDRESS P.AIA
00 20 65 6D 61 6E 20 65 67 61 6D 49 01DD8 P.AIC: .ASCII \Image name \<0>
00000008' 01DE4 P.AIB: .LONG 11
00000000' 01DE8 .ADDRESS P.AIC
00000000' 01DEC P.AHY: .ADDRESS IMAGE DESC, P.AHZ, P.AIB
00 00 00 53 41 39 33 21 01DF8 P.AID: .ASCII <3>\JOB\
00000005' 01DFC P.AIG: .ASCII \!39AS\<0><0><0>
00000000' 01E04 P.AIF: .LONG 5
00000000' 01E08 .ADDRESS P.AIG
20 20 20 20 20 20 20 65 6D 61 6E 20 62 6F 4A 01E0C P.AII: .ASCII \Job name
20 20 20 20 20 20 20 20 20 20 20 20 20 20 01E1B \<0>
20 20 20 20 20 20 20 20 20 20 20 20 20 20 01E2A
00000027' 01E34 P.AIH: .LONG 39
00000000' 01E38 .ADDRESS P.AII
00000000' 01E3C P.AIE: .ADDRESS JOB DESC, P.AIF, P.AIH
00 00 48 54 4E 4F 4D 05 01E48 P.AIJ: .ASCII <5>\MONTH\<0><0>
53 41 33 21 01E50 P.AIM: .ASCII \!3AS\
00000004' 01E54 P.AIL: .LONG 4
00000000' 01E58 .ADDRESS P.AIM
00 20 4D 4D 01E5C P.AIO: .ASCII \MM \<0>
00000003' 01E60 P.AIN: .LONG 3
00000000' 01E64 .ADDRESS P.AIO
00000000' 01E68 P.AIK: .ADDRESS MONTH DESC, P.AIL, P.AIN
00 00 00 45 44 4F 4E 04 01E74 P.AIP: .ASCII <4>\NODE\<0><0><0>
00 00 00 53 41 30 31 21 01E7C P.AIS: .ASCII \!10AS\<0><0><0>
00000005' 01E84 P.AIR: .LONG 5
00000000' 01E88 .ADDRESS P.AIS
00 00 20 65 6D 61 6E 20 65 64 6F 4E 01E8C P.AIU: .ASCII \Node name \<0><0>
0000000A' 01E98 P.AIT: .LONG 10
00000000' 01E9C .ADDRESS P.AIU
00000000' 01EA0 P.AIQ: .ADDRESS NODE DESC, P.AIR, P.AIT
53 53 45 43 4F 52 50 07 01EAC P.AIV: .ASCII <7>\PROCESS\
00 00 00 53 41 33 31 21 01EB4 P.AIY: .ASCII \!13AS\<0><0><0>
00000005' 01EBC P.AIX: .LONG 5
00000000' 01EC0 .ADDRESS P.AIY
00 00 20 65 70 79 74 20 73 73 65 63 6F 72 50 01EC4 P.AJA: .ASCII \Process type \<0><0><0>
00 00 00 00 01ED3
00000000' 01ED4 P.AIZ: .LONG 13
00000000' 01ED8 .ADDRESS P.AJA
00000000' 01EDC P.AIW: .ADDRESS PROCESS DESC, P.AIX, P.AIZ
00 00 45 55 45 55 51 05 01EE8 P.AJB: .ASCII <5>\QUEUE\<0><0>
```



```
00 00 00 53 41 31 33 21 01EF0 P.AJE: .ASCII \!31AS\<0><0><0>
00000005 01EF8 P.AJD: .LONG 5
00000000 01EFC P.AJE: .ADDRESS P.AJE
65 75 65 75 71 20 74 6E 69 72 70 2F 62 6F 4A 01F00 P.AJG: .ASCII \Job/print queue \<0>
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 01F0F
00 00 01F1E
0000001F 01F20 P.AJF: .LONG 31
00000000 01F24 P.AJG: .ADDRESS P.AJG
00000000 01F28 P.AJC: .ADDRESS QUEUE DESC, P.AJD, P.AJF
00 00 00 4C 41 4E 49 4D 52 45 54 08 01F34 P.AJH: .ASCII <8>\TERMINAL\<0><0><0>
53 41 39 21 01F40 P.AJK: .ASCII \!9AS\
00000004 01F44 P.AJJ: .LONG 4
00000000 01F48 P.AJK: .ADDRESS P.AJK
00 00 00 20 6C 61 6E 69 6D 72 65 54 01F4C P.AJM: .ASCII \Terminal \<0><0><0>
00000009 01F58 P.AJL: .LONG 9
00000000 01F5C P.AJM: .ADDRESS P.AJM
00000000 01F60 P.AJI: .ADDRESS TERMINAL DESC, P.AJJ, P.AJL
00 00 00 45 50 59 54 04 01F6C P.AJN: .ASCII <4>\TYPE\<0><0><0>
53 41 38 21 01F74 P.AJQ: .ASCII \!8AS\
00000004 01F78 P.AJP: .LONG 4
00000000 01F7C P.AJQ: .ADDRESS P.AJQ
20 20 20 20 65 70 79 54 01F80 P.AJS: .ASCII \Type \
00000008 01F88 P.AJR: .LONG 8
00000000 01F8C P.AJS: .ADDRESS P.AJS
00000000 01F90 P.AJO: .ADDRESS TYPE DESC, P.AJP, P.AJR
00 00 00 53 41 37 36 21 01F9C P.AJT: .ASCII <3>\UIC\
00000005 01FA0 P.AJW: .ASCII \!67AS\<0><0><0>
00000000 01FAB P.AJV: .LONG 5
00000000 01FAC P.AJW: .ADDRESS P.AJW
20 20 20 20 20 20 20 43 20 20 49 20 20 55 20 01FB0 P.AJY: .ASCII \ U I C \
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 01FBF
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 01FCE
00 20 20 20 20 20 20 20 20 20 20 20 20 20 20 01FD8
00 20 20 20 20 20 20 20 20 20 20 20 20 20 20 01FE7
00000043 01FF4 P.AJX: .LONG 67
00000000 01FF8 P.AJY: .ADDRESS P.AJY
00000000 01FFC P.AJU: .ADDRESS UIC DESC, P.AJV, P.AJX
00 00 00 52 45 53 55 04 02008 P.AJZ: .ASCII <4>\USER\<0><0><0>
00 00 00 53 41 33 31 21 02010 P.AKC: .ASCII \!13AS\<0><0><0>
00000005 02018 P.AKB: .LONG 5
00000000 0201C P.AKC: .ADDRESS P.AKC
00 00 20 20 20 20 20 65 6D 61 6E 72 65 73 55 02020 P.AKE: .ASCII \Username \<0><0><0>
00 0202F
00000000 02030 P.AKD: .LONG 13
00000000 02034 P.AKE: .ADDRESS P.AKE
00000000 02038 P.AKA: .ADDRESS USER DESC, P.AKB, P.AKD
59 41 44 48 45 45 57 07 02044 P.AKF: .ASCII <7>\WEEKDAY\
53 41 33 21 0204C P.AKI: .ASCII \!3AS\
00000004 02050 P.AKH: .LONG 4
00000000 02054 P.AKI: .ADDRESS P.AKI
00 20 44 44 02058 P.AKK: .ASCII \DD \<0>
00000003 0205C P.AKJ: .LONG 3
00000000 02060 P.AKK: .ADDRESS P.AKK
00000000 02064 P.AKG: .ADDRESS WEEKDAY DESC, P.AKH, P.AKJ
00 00 00 52 41 45 59 04 02070 P.AKL: .ASCII <4>\YEAR\<0><0><0>
53 41 35 21 02078 P.AKO: .ASCII \!5AS\
00000004 0207C P.AKN: .LONG 4
```

				00	00	00	20	72	61	65	59	00000000	02080		.ADDRESS P.AKQ	
												00000005	02084	P.AKQ:	.ASCII \Year \<0><0><0>	
												00000000	0208C	P.AKP	.LONG 5	
												00000000	02090		.ADDRESS P.AKQ	
				00000000		00000000						00000000	02094	P.AKM:	.ADDRESS YEAR DESC, P.AKN, P.AKP	
4F	49	5F	44	45	52	45	46	46	55	42	0B	00000000	020A0	P.AKR:	.ASCII <11>\BUFFERED IO\	
				00	00	00	4C	55	38	21	20	00000005	020AC	P.AKJ:	.ASCII \!8UL\<0><0><20>	
												00000000	020B4	P.AKT:	.LONG 5	
												00000000	020B8		.ADDRESS P.AKU	
00	00	00	64	65	72	65	66	66	75	42	20	00000009	020BC	P.AKW:	.ASCII \ Buffered\<0><0><0>	
												00000000	020C8	P.AKV:	.LONG 9	
00	00	00	20	20	4F	2F	49	20	20	20	20	00000000	020CC		.ADDRESS P.AKW	
												00000009	020D0	P.AKY:	.ASCII \ I/O \<0><0><0>	
												00000000	020DC	P.AKX:	.LONG 9	
												00000000	020E0		.ADDRESS P.AKY	
												00000000	020E4	P.AKS:	.LONG 0	
												00000000	020E8		.ADDRESS HOLD_BIOCNT	
												00000001	020EC		.LONG 1	
				00000000		00000000						00000000	020F0		.ADDRESS P.AKT, P.AKV, P.AKX	
00	00	4F	49	5F	54	43	45	52	49	44	09	00000000	020FC	P.AKZ:	.ASCII <9>\DIRECT IO\<0><0>	
				00	00	00	4C	55	38	21	20	00000005	02108	P.ALC:	.ASCII \!8UL\<0><20><0>	
												00000000	02110	P.ALB:	.LONG 5	
												00000000	02114		.ADDRESS P.ALC	
00	00	00	74	63	65	72	69	44	20	20	20	00000009	02118	P.ALE:	.ASCII \ Direct\<0><0><0>	
												00000000	02124	P.ALD:	.LONG 9	
												00000000	02128		.ADDRESS P.ALE	
00	00	00	20	20	4F	2F	49	20	20	20	20	00000009	0212C	P.ALG:	.ASCII \ I/O \<0><0><0>	
												00000000	02138	P.ALF:	.LONG 9	
												00000000	0213C		.ADDRESS P.ALG	
												00000000	02140	P.ALA:	.LONG 0	
												00000000	02144		.ADDRESS HOLD_DIOCNT	
												00000001	02148		.LONG 1	
				00000000		00000000						00000000	0214C		.ADDRESS P.ALB, P.ALD, P.ALF	
00	00	00	54	44	45	53	50	41	4C	45	07	00000000	02158	P.ALH:	.ASCII <7>\ELAPSED\	
				25	21	20	4C	55	34	21	20	00000009	02160	P.ALK:	.ASCII \!4UL!XT\<0><0><0>	
												00000000	0216C	P.ALJ:	.LONG 9	
												00000000	02170		.ADDRESS P.ALK	
20	20	20	64	65	73	70	61	6C	45	20	20	20	02174	P.ALM:	.ASCII \ Elapsed \<0><0><0>	
										00	00	00	02183			
												00000011	02188	P.ALL:	.LONG 17	
												00000000	0218C		.ADDRESS P.ALM	
20	20	20	20	65	6D	69	54	20	20	20	20	20	02190	P.ALO:	.ASCII \ Time \<0><0><0>	
										00	00	00	0219F			
												00000011	021A4	P.ALN:	.LONG 17	
												00000000	021A8		.ADDRESS P.ALO	
												00000001	021AC	P.ALI:	.LONG 1	
												00000000	021B0		.ADDRESS HOLD_ELAPSED	
												00000004	021B4		.LONG 4	
				00000000		00000000						00000000	021B8		.ADDRESS P.ALJ, P.ALL, P.ALN	
00	00	4E	4F	49	54	55	43	45	58	45	09	00000000	021C4	P.ALP:	.ASCII <9>\EXECUTION\<0><0>	
				00	00	00	4C	55	38	21	20	00000009	021D0	P.ALS:	.ASCII \!8UL\<0><0><0>	
												000CJ005	021D8	P.ALR:	.LONG 5	
												00000000	021DC		.ADDRESS P.ALS	
00	00	00	20	20	65	67	61	6D	49	20	20	00000009	021E0	P.ALU:	.ASCII \ Image \<0><0><0>	
												00000000	021EC	P.ALT:	.LONG 9	
												00000000	021F0		.ADDRESS P.ALU	
00	00	00	20	74	6E	75	6F	43	20	20	20	00000000	021F4	P.ALW:	.ASCII \ Count \<0><0><0>	

```
000000009 02200 P.ALX: .LONG 9
000000000 02204 .ADDRESS P.ALW
000000000 02208 P.ALQ: .LONG 0
000000000 0220C .ADDRESS HOLD_EXECUTION
000000001 02210 .LONG 1
000000000 02214 .ADDRESS P.ALR, P.ALT, P.ALV
00 53 54 4C 55 41 46 06 02220 P.ALX: .ASCII <6>\FAULTS\<0>
00 00 00 4C 55 38 21 20 02228 P.AMA: .ASCII \!8UL\<0><0><0>
000000005 02230 P.ALZ: .LONG 5
000000000 02234 .ADDRESS P.AMA
00 00 00 20 65 67 61 5J 20 20 20 02238 P.AMC: .ASCII \ Page \<0><0><0>
000000009 02244 P.AMB: .LONG 9
000000000 02248 .ADDRESS P.AMC
00 00 00 73 74 6C 75 61 46 20 20 20 0224C P.AME: .ASCII \ Faults\<0><0><0>
000000009 02258 P.AMD: .LONG 9
000000000 0225C .ADDRESS P.AME
000000000 02260 P.ALY: .LONG 0
000000000 02264 .ADDRESS HOLD_PAGEFLTS
000000001 02268 .LONG 1
000000000 0226C .ADDRESS P.ALZ, P.AMB, P.AMD
00 00 00 53 54 45 47 04 02278 P.AMF: .ASCII <4>\GETS\<0><0><0>
00 00 00 4C 55 38 21 20 02280 P.AMI: .ASCII \!8UL\<0><0><0>
000000005 02288 P.AMH: .LONG 5
000000000 0228C .ADDRESS P.AMI
00 00 00 74 6E 6F 69 62 6D 79 53 20 02290 P.AMK: .ASCII \ Symbiont\<0><0><0>
000000009 0229C P.AMJ: .LONG 9
000000000 022A0 .ADDRESS P.AMK
00 00 00 20 20 73 74 65 47 20 20 20 022A4 P.AMM: .ASCII \ Gets \<0><0><0>
000000009 022B0 P.AML: .LONG 9
000000000 022B4 .ADDRESS P.AMM
000000000 022B8 P.AMG: .LONG 0
000000000 022BC .ADDRESS HOLD_GETCNT
000000001 022C0 .LONG 1
000000000 022C4 .ADDRESS P.AMH, P.AMJ, P.AML
00 00 53 45 47 41 50 05 022D0 P.AMN: .ASCII <5>\PAGES\<0><0>
00 00 00 4C 55 38 21 20 022D8 P.AMQ: .ASCII \!8UL\<0><0><0>
000000005 022E0 P.AMP: .LONG 5
000000000 022E4 .ADDRESS P.AMQ
00 00 00 73 65 67 61 50 20 20 20 022E8 P.AMS: .ASCII \ Pages\<0><0><0>
000000009 022F4 P.AMR: .LONG 9
000000000 022F8 .ADDRESS P.AMS
00 00 00 64 65 74 6E 69 72 50 20 20 022FC P.AMU: .ASCII \ Printed\<0><0><0>
000000009 02308 P.AMT: .LONG 9
000000000 0230C .ADDRESS P.AMU
000000000 02310 P.AMO: .LONG 0
000000000 02314 .ADDRESS HOLD_PAGCNT
000000001 02318 .LONG 1
000000000 0231C .ADDRESS P.AMP, P.AMR, P.AMT
00 00 45 4C 49 46 5F 45 47 41 50 09 02328 P.AMV: .ASCII <9>\PAGE FILE\<0><0>
00 00 00 4C 55 38 21 20 02334 P.AMY: .ASCII \!8UL\<0><0><0>
000000005 0233C P.AMX: .LONG 5
000000000 02340 .ADDRESS P.AMY
00 00 00 65 67 61 70 68 61 65 50 20 02344 P.ANA: .ASCII \ Peakpage\<0><0><0>
000000009 02350 P.AMZ: .LONG 9
000000000 02354 .ADDRESS P.ANA
00 00 00 65 73 75 20 65 6C 69 46 20 02358 P.ANC: .ASCII \ File use\<0><0><0>
000000009 02364 P.ANB: .LONG 9
```

```
00000000' 02368 .ADDRESS P.ANC
00000002' 0236C P.AMW: .LONG 2
00000000' 02370 .ADDRESS HOLD_PGFLPEAK
00000001' 02374 .LONG 1
00000000' 02378 .ADDRESS P.AMX, P.AMZ, P.ANB
00 53 44 41 45 52 5F 45 47 41 50 0A 02384 P.AND: .ASCII <10>\PAGE READS\<0>
00 00 00 4C 55 38 21 20 02390 P.ANG: .ASCII \!8UL\<0>\<0>\<0>
00000005' 02398 P.ANF: .LONG 5
00000000' 0239C .ADDRESS P.ANG
00 00 00 20 4F 49 20 65 67 61 50 20 023A0 P.ANI: .ASCII \ Page 10 \<0>\<0>\<0>
00000009' 023AC P.ANH: .LONG 9
0000000C' 023B0 .ADDRESS P.ANI
00 00 00 20 20 20 73 64 61 65 52 20 023B4 P.ANK: .ASCII \ Reads \<0>\<0>\<0>
00000009' 023C0 P.ANJ: .LONG 9
00000000' 023C4 .ADDRESS P.ANK
00000000' 023C8 P.ANE: .LONG 0
00000000' 023CC .ADDRESS HOLD_PAGEREADS
00000001' 023D0 .LONG 1
00000000' 023D4 .ADDRESS P.ANF, P.ANH, P.ANJ
00 00 52 4F 53 53 45 43 4F 52 50 09 023E0 P.ANL: .ASCII <9>\PROCESSOR\<0>\<0>
00 00 00 54 25 21 20 4C 55 34 21 20 023EC P.ANO: .ASCII \!4UL!XT\<0>\<0>\<0>
00000009' 023F8 P.ANN: .LONG 9
00000000' 023FC .ADDRESS P.ANO
20 20 72 6F 73 73 65 63 6F 72 50 20 20 20 20 02400 P.ANQ: .ASCII \ Processor \<0>\<0>\<0>
00 00 00 20 20 20
00000011' 02414 P.ANP: .LONG 17
00000000' 02418 .ADDRESS P.ANQ
20 20 20 20 65 6D 69 54 20 20 20 20 20 20 20 0241C P.ANS: .ASCII \ Time \<0>\<0>\<0>
00 00 00 20 20 20
00000011' 02430 P.ANR: .LONG 17
00000000' 02434 .ADDRESS P.ANS
00000001' 02438 P.ANM: .LONG 1
00000000' 0243C .ADDRESS HOLD_CPUIM
00000004' 02440 .LONG 4
00000000' 02444 .ADDRESS P.ANN, P.ANP, P.ANR
00 00 00 53 4F 49 51 04 02450 P.ANT: .ASCII <4>\QIOS\<0>\<0>\<0>
00 00 00 4C 55 38 21 20 02458 P.ANW: .ASCII \!8UL\<0>\<0>\<0>
00000005' 02460 P.ANV: .LONG 5
00000000' 02464 .ADDRESS P.ANW
00 00 00 74 6E 6F 69 62 6D 79 53 20 02468 P.ANY: .ASCII \ Symbiont\<0>\<0>\<0>
00000009' 02474 P.ANX: .LONG 9
00000000' 02478 .ADDRESS P.ANY
00 00 00 20 73 27 4F 49 51 20 20 20 0247C P.AOA: .ASCII \ QIO's \<0>\<0>\<0>
00000009' 02488 P.ANZ: .LONG 9
00000000' 0248C .ADDRESS P.AOA
00000000' 02490 P.ANU: .LONG 0
00000000' 02494 .ADDRESS HOLD_QIOCNT
00000001' 02498 .LONG 1
00000000' 0249C .ADDRESS P.ANV, P.ANX, P.ANZ
53 44 52 4F 43 45 52 07 024A8 P.AOB: .ASCII <7>\RECORDS\
00 00 00 4C 55 38 21 20 024B0 P.AOE: .ASCII \!8UL\<0>\<0>\<0>
00000005' 024B8 P.AOD: .LONG 5
00000000' 024BC .ADDRESS P.AOE
00 00 00 6C 61 74 6F 54 20 20 20 20 024C0 P.AOG: .ASCII \ Total\<0>\<0>\<0>
00000009' 024CC P.AOF: .LONG 9
00000000' 024D0 .ADDRESS P.AOG
00 00 00 73 64 72 6F 63 65 52 20 20 024D4 P.AOI: .ASCII \ Records\<0>\<0>\<0>
```

```
00000009 024E0 P.AOH: .LONG 9
00000000 024E4 .ADDRESS P.AOI
00000003 024E8 P.AOC: .LONG 3, 0, 1
00000000 024F4 .ADDRESS P.AOD, P.AOF, P.AOH
53 45 4D 55 4C 4F 56 07 02500 P.AOJ: .ASCII <7>\VOLUMES\
00 00 00 4C 55 38 21 20 02508 P.AOM: .ASCII \ !8UL\<0><0><0>
00000005 02510 P.AOL: .LONG 5
00000000 02514 .ADDRESS P.AOM
00 00 00 73 65 6D 75 6C 6F 56 20 20 02518 P.AOO: .ASCII \ Volumes\<0><0><0>
00000009 02524 P.AON: .LONG 9
00000000 02528 .ADDRESS P.AOO
00 00 00 64 65 74 6E 75 6F 4D 20 20 0252C P.AOQ: .ASCII \ Mounted\<0><0><0>
00000009 02538 P.AOP: .LONG 9
00000000 0253C .ADDRESS P.AOQ
00000000 02540 P.AOK: .LONG 0
00000000 02544 .ADDRESS HOLD_VOLUMES
00000001 02548 .LONG 1
00000000 0254C .ADDRESS P.AOL, P.AON, P.AOP
54 45 53 5F 47 4E 49 48 52 4F 57 0B 02558 P.AOR: .ASCII <11>\WORKING_SET\
00 00 00 4C 55 38 21 20 02564 P.AOU: .ASCII \ !8UL\<0><0><0>
00000005 0256C P.AOT: .LONG 5
00000000 02570 .ADDRESS P.AOU
00 00 00 68 72 6F 77 6B 61 65 50 20 02574 P.AOW: .ASCII \ Peakwork\<0><0><0>
00000009 02580 P.AOV: .LONG 9
00000000 02584 .ADDRESS P.AOW
00 00 00 65 67 61 70 20 74 65 53 20 02588 P.AOY: .ASCII \ Set page\<0><0><0>
00000009 02594 P.AOX: .LONG 9
00000000 02598 .ADDRESS P.AOY
00000002 0259C P.AOS: .LONG 2
00000000 025A0 .ADDRESS HOLD_WSPEAK
00000001 025A4 .LONG 1
00000000 025A8 .ADDRESS P.AOT, P.AOV, P.AOX
00000000 025B4 P.APA: .ASCII \SORT\
54 52 4F 53 025B8 P.AOZ: .LONG 4
00000004 025BC .ADDRESS P.APA
00 59 52 41 4D 4D 55 53 025C0 P.APC: .ASCII \SUMMARY\<0>
00000007 025C8 P.APB: .LONG 7
00000000 025CC .ADDRESS P.APC
52 45 53 55 025D0 P.APE: .ASCII \USER\
00000004 025D4 P.APD: .LONG 4
00000000 025D8 .ADDRESS P.APE
00 4C 49 41 46 47 4F 4C 025DC P.APG: .ASCII \LOGFAIL\<0>
00000007 025E4 P.APF: .LONG 7
00000000 025E8 .ADDRESS P.APG
00 43 49 55 025EC P.API: .ASCII \UIC\<0>
00000003 025F0 P.APH: .LONG 3
00000000 025F4 .ADDRESS P.API
52 55 4F 48 025F8 P.APK: .ASCII \HOUR\
00000004 025FC P.APJ: .LONG 4
00000000 02600 .ADDRESS P.APK
00 59 41 44 4B 45 45 57 02604 P.APM: .ASCII \WEEKDAY\<0>
00000007 0260C P.APL: .LONG 7
00000000 02610 .ADDRESS P.APM
00 59 41 44 02614 P.APO: .ASCII \DAY\<0>
00000003 02618 P.APN: .LONG 3
00000000 0261C .ADDRESS P.APO
00 00 00 48 54 4E 4F 4D 02620 P.APQ: .ASCII \MONTH\<0><0><0>
```

```
00000005 02628 P.APP: .LONG 5
00000000 0262C .ADDRESS P.APO
52 41 45 59 02630 P.APS: .ASCII \YEAR\
00000004 02634 P.APR: .LONG 4
00000000 02638 .ADDRESS P.APS
45 54 41 44 0263C P.APU: .ASCII \DATE\
00000004 02640 P.APT: .LONG 4
00000000 02644 .ADDRESS P.APU
00 53 41 21 02648 P.APW: .ASCII \AS\<0>
00000003 0264C P.APV: .LONG 3
00000000 02650 .ADDRESS P.APW
00 00 54 52 4F 50 45 52 02654 P.APY: .ASCII \REPORT\<0><0>
00000006 0265C P.APX: .LONG 6
00000000 02660 .ADDRESS P.APY
```

.PSECT \$OWNS,NOEXE,2

```
00# 00110 DESC: .BYTE 0[3]
02 00113 .BYTE 2
00114 .BLKB 4
00000040 00118 SORT_KEYWORD:
00000000 0011C .LONG 64
00000000 00120 .ADDRESS P.AFT
00000000 00124 .LONG 0
00000001 00128 .ADDRESS P.AFU
00000000 0012C .LONG 1
00000002 00130 .ADDRESS P.AFV
00000000 00134 .LONG 2
00000003 00138 .ADDRESS P.AFW
00000000 0013C .LONG 3
00000004 00140 .ADDRESS P.AFX
00000000 00144 .LONG 4
00000005 00148 .ADDRESS P.AFY
00000000 0014C .LONG 5
00000006 00150 .ADDRESS P.AFZ
00000007 00154 .LONG 6
00000000 00158 .ADDRESS P.AGA
00000000 00160 .LONG 7
00000008 00164 .ADDRESS P.AGB
00000009 00168 .LONG 8
00000000 0016C .ADDRESS P.AGC
0000000A 00170 .LONG 9
00000000 00174 .ADDRESS P.AGD
0000000B 00178 .LONG 10
00000000 0017C .ADDRESS P.AGE
0000000C 00180 .LONG 11
00000000 00184 .ADDRESS P.AGF
0000000D 00188 .LONG 12
0000000E 00190 .ADDRESS P.AGG
0000000F 00194 .LONG 13
00000000 00198 .ADDRESS P.AGH
00000000 0019C .LONG 14
00000010 001A0 .ADDRESS P.AGI
00000000 001A4 .LONG 15
00000000 001A8 .ADDRESS P.AGJ
00000000 001AC .LONG 16
00000000 001B0 .ADDRESS P.AGK
```

00000011 001A8 .LONG 17
00000000 001AC .ADDRESS P.AGL
00000012 001B0 .LONG 18
00000000 001B4 .ADDRESS P.AGM
00000013 001B8 .LONG 19
00000000 001BC .ADDRESS P.AGN
00000014 001C0 .LONG 20
00000000 001C4 .ADDRESS P.AGO
00000015 001C8 .LONG 21
00000000 001CC .ADDRESS P.AGP
00000016 001D0 .LONG 22
00000000 001D4 .ADDRESS P.AGQ
00000017 001D8 .LONG 23
00000000 001DC .ADDRESS P.AGR
00000018 001E0 .LONG 24
00000000 001E4 .ADDRESS P.AGS
00000019 001E8 .LONG 25
00000000 001EC .ADDRESS P.AGT
0000001A 001F0 .LONG 26
00000000 001F4 .ADDRESS P.AGU
0000001B 001F8 .LONG 27
00000000 001FC .ADDRESS P.AGV
0000001C 00200 .LONG 28
00000000 00204 .ADDRESS P.AGW
0000001D 00208 .LONG 29
00000000 0020C .ADDRESS P.AGX
0000001E 00210 .LONG 30
00000000 00214 .ADDRESS P.AGY
0000001F 00218 .LONG 31
00000020 0021C SUMMARY_KEYWORD:

00000000 00000000 00000000 00000000 00000000 00000000 00220 .LONG 32
00000000 00000000 00000000 00000000 00000000 00000000 00238 .ADDRESS P.AGZ, P.AHA, P.AHF, P.AHG, P.AHL, -
00000000 00000000 00000000 00000000 00000000 00000000 00250 P.AHM, P.AHR, P.AHS, P.AHX, P.AHY, P.AID, -
00000000 00000000 00000000 00000000 00000000 00000000 00268 P.AIE, P.AIJ, P.AIK, P.AIP, P.AIQ, P.AIV, -
00000000 00000000 00000000 00000000 00000000 00000000 00280 P.AIW, P.AJB, P.AJC, P.AJH, P.AJI, P.AJN, -
00000000 00000000 00000000 00000000 00000000 00000000 00298 P.AJO, P.AJT, P.AJU, P.AJZ, P.AKA, P.AKF, -
0000001C 002A0 P.AKG, P.AKL, P.AKM

00000000 00000000 00000000 00000000 00000000 00000000 002A4 .LONG 28
00000000 00000000 00000000 00000000 00000000 00000000 002BC .ADDRESS P.AKR, P.AKS, P.AKZ, P.ALA, P.ALH, -
00000000 00000000 00000000 00000000 00000000 00000000 002D4 P.ALI, P.ALP, P.ALQ, P.ALX, P.ALY, P.AMF, -
00000000 00000000 00000000 00000000 00000000 00000000 002EC P.AMG, P.AMN, P.AMO, P.AMV, P.AMW, P.AND, -
00000000 00000000 00000000 00000000 00000000 00000000 00304 P.ANE, P.ANL, P.ANM, P.ANT, P.ANU, P.AOB, -
P.AOC, P.AOJ, P.AOK, P.AOR, P.AOS

.PSECT \$GLOBALS,NOEXE,2

00000001 00000 KEY_TABLE::
00000000 00004 .LONG 1
00000002 00008 .ADDRESS ACCOUNT_DESC
00000000 00010 .LONG 8, 2
00000002 00014 .ADDRESS ENTRY_DESC
00000000 0001C .LONG 4, 2
00000002 00020 .ADDRESS FINISH_TIME_DESC
00000000 00028 .LONG 8, 2
00000001 0002C .ADDRESS IDENT_DESC
00000004 0002C .LONG 4, 1

	00000000	00034	.ADDRESS IMAGE_DESC
00000001	00000009	00038	.LONG 9, 1
	00000000	00040	.ADDRESS JOB_DESC
00000001	00000027	00044	.LONG 39, 1
	00000000	0004C	.ADDRESS NODE_DESC
00000002	00000006	00050	.LONG 6, 2
	00000000	00058	.ADDRESS OWNER_DESC
00000001	00000004	0005C	.LONG 4, 1
	00000000	00064	.ADDRESS PROCESS_DESC
00000001	0000000B	00068	.LONG 11, 1
	00000000	00070	.ADDRESS QUEUE_DESC
00000002	0000001F	00074	.LONG 31, 2
	00000000	0007C	.ADDRESS QUEUE_TIME_DESC
00000002	00000008	00080	.LONG 8, 2
	00000000	00088	.ADDRESS START_TIME_DESC
00000002	00000008	0008C	.LONG 8, 2
	00000000	00094	.ADDRESS STATUS_DESC
00000001	00000004	00098	.LONG 4, 1
	00000000	000A0	.ADDRESS TERMINAL_DESC
00000001	00000008	000A4	.LONG 8, 1
	00000000	000AC	.ADDRESS TYPE_DESC
00000001	00000008	000B0	.LONG 8, 1
	00000000	000B8	.ADDRESS USER_DESC
00000002	0000000C	000BC	.LONG 12, 2
	00000000	000C4	.ADDRESS ADDRESS_DESC
00000002	00000004	000C8	.LONG 4, 2
	00000000	000D0	.ADDRESS BUFFERED_IO_DESC
00000002	00000004	000D4	.LONG 4, 2
	00000000	000DC	.ADDRESS DIRECT_IO_DESC
00000002	00000004	000E0	.LONG 4, 2
	00000000	000E8	.ADDRESS CPU_TIME_DESC
00000002	00000004	000EC	.LONG 4, 2
	00000000	000F4	.ADDRESS PAGE_FAULTS_DESC
00000002	00000004	000F8	.LONG 4, 2
	00000000	00100	.ADDRESS PAGE_READS_DESC
00000002	00000004	00104	.LONG 4, 2
	00000000	0010C	.ADDRESS PAGE_FILE_DESC
00000002	00000004	00110	.LONG 4, 2
	00000000	00118	.ADDRESS WORKING_SET_DESC
00000002	00000004	0011C	.LONG 4, 2
	00000000	00124	.ADDRESS EXECUTION_DESC
00000002	00000004	00128	.LONG 4, 2
	00000000	00130	.ADDRESS ELAPSED_DESC
00000002	00000008	00134	.LONG 8, 2
	00000000	0013C	.ADDRESS PRIORITY_DESC
00000002	00000001	00140	.LONG 1, 2
	00000000	00148	.ADDRESS VOLUMES_DESC
00000002	00000004	0014C	.LONG 4, 2
	00000000	00154	.ADDRESS SYM_PAGE_DESC
00000002	00000004	00158	.LONG 4, 2
	00000000	00160	.ADDRESS SYM_GET_DESC
00000002	00000004	00164	.LONG 4, 2
	00000000	0016C	.ADDRESS SYM_QIO_DESC
00000002	00000004	00170	.LONG 4, 2
	00000000	00178	.ADDRESS UIC_BINARY_DESC
	00000004	0017C	.LONG 4


```
.PSECT CODE,NOWRT,2

OFFC C0000 PARSE_KEYS:
        .WORD      Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11
5B 00030000G 00 9E 00002 MOVAB QUALIFIERS, R11
5A 00000C00G 00 9E 00009 MOVAB LIB$LOOKUP_KEY, R10
59 00000000G 00 9E 00010 MOVAB CLISGET_VALUE, R9
58 FF39 CF 9E 00017 MOVAB P.AOZ, R8
57 00000000G 00 9E 0001C MOVAB STR$APPEND, R7
56 00000000G 00 9E 00023 MOVAB STR$COMPARE, R6
55 0000' CF 9E 0002A MOVAB DESC, R5
54 00000000' EF 9E 0002F MOVAB BUCKET_SIZE, R4
5E 00A8 0C C2 00036 SUBL2 #12, SP
        00A8 C4 B4 00039 CLRW SORT_KEY_COUNT
        53 D4 0003D CLRL I
        55 DD 0003F 1$: PUSHL R5
        58 DD 00041 PUSHL R8
        69 02 FB 00043 CALLS #2, CLISGET_VALUE
        75 50 E9 00046 BLBC R0, 4$
        52 D4 00049 CLRL KEY_ORDER
        55 DD 0004B PUSHL R5
00000000G 00 01 FB 0004D CALLS #1, STRIP_NEGATOR
03 50 E8 00054 BLBS R0, 2$
52 01 D0 00057 MOVL #1, KEY_ORDER
        04 AE 9F 0005A 2$: PUSHAB INDEX
        08 A5 9F 0005D PUSHAB SORT_KEYWORD
        55 DD 00060 PUSHL R5
        6A 03 FB 00062 CALLS #3, LIB$LOOKUP_KEY
        03 50 E8 00065 BLBS STATUS, 3$
        016C 31 00068 BRW 14$
51 04 50 00A8 04 3C 0006B 3$: MOVZWL SORT_KEY_COUNT, R0
        AE 0C C5 00070 MULL3 #12, INDEX, R1
        00AC C440 7F 00075 PUSHAB SORT_TABLE[R0]
        0000' CF41 9F 0007A PUSHAB KEY_TABLE[R1]
        9E 9E B0 0007F MOVW @ (SP)+, @ (SP)+
        00AE C440 7F 00082 PUSHAB SORT_TABLE+2[R0]
        9E 52 B0 00087 MOVW KEY_ORDER, @ (SP)+
        00B0 C440 7F 0008A PUSHAB SORT_TABLE+4[R0]
9E FE50 C4 01 A1 0008F ADDW3 #1, SORT_KEY_SIZE, @ (SP)+
        00B2 C440 7F 00095 PUSHAB SORT_TABLE+6[R0]
        0000' CF41 9F 0009A PUSHAB KEY_TABLE+8[R1]
        9E 9E B0 0009F MOVW @ (SP)+, @ (SP)+
        0104 C440 04 AE 90 000A2 MOVAB INDEX, SORT_INDEX_TABLE[R0]
        0000' CF41 9F 000A9 PUSHAB KEY_TABLE+8[R1]
        50 9E 3C 000AE MOVZWL @ (SP)+, R0
        FE50 C4 50 C0 000B1 ADDL2 R0, SORT_KEY_SIZE
        00A8 C4 B6 000B6 INCW SORT_KEY_COUNT
81 53 0A F3 000BA AOBLEQ #10, I, T$
        53 D4 000BE 4$: CLRL I
        55 DD 000C0 5$: PUSHL R5
        10 A8 9F 000C2 PUSHAB P.APB
        69 02 FB 000C5 CALLS #2, CLISGET_VALUE
        03 50 E8 000C8 BLBS R0, 6$
        00C9 31 000CB BRW 11$
        1C A8 9F 000CE 6$: PUSHAB P.APD
```

			55	DD	000D1	PUSHL	R5		
		66	02	FB	000D3	CALLS	#2, STR\$COMPARE		
			50	D5	000D6	TSTL	R0		
			1B	12	000D8	BNEQ	7\$		
16	03	AB	03	E0	000DA	BBS	#3, QUALIFIERS+3, 7\$		3781
	03	AB	08	88	000DF	BISB2	#8, QUALIFIERS+3		3784
			01	DD	000E3	PUSHL	#1		3785
			2C	A8	9F	000E5	PUSHAB	P.APF	
			1B	DD	000E8	PUSHL	#27		
00000000G		00	03	FB	000EA	CALLS	#3, ADD_SYMBOL		
		U1	50	E8	000F1	BLBS	STATUS, -7\$		
				04	000F4	RET			
			38	A8	9F	000F5	7\$: PUSHAB	P.APH	3788
			55	DD	000F8	PUSHL	R5		
		66	02	FB	000FA	CALLS	#2, STR\$COMPARE		
			50	D5	000FD	TSTL	R0		
			05	12	000FF	BNEQ	8\$		
0110		C4	01	DD	00101	MOVL	#1, SUMMARY_UIC		3790
			44	A8	9F	00106	8\$: PUSHAB	P.APJ	3792
			55	DD	00109	PUSHL	R5		
		66	02	FB	0010B	CALLS	#2, STR\$COMPARE		
			50	D5	0010E	TSTL	R0		
			3D	13	00110	BEQL	9\$		
			54	A8	9F	00112	PUSHAB	P.APL	3793
			55	DD	00115	PUSHL	R5		
		66	02	FB	00117	CALLS	#2, STR\$COMPARE		
			50	D5	0011A	TSTL	R0		
			31	13	0011C	BEQL	9\$		
			60	A8	9F	0011E	PUSHAB	P.APN	3794
			55	DD	00121	PUSHL	R5		
		66	02	FB	00123	CALLS	#2, STR\$COMPARE		
			50	D5	00126	TSTL	R0		
			25	13	00128	BEQL	9\$		
			70	A8	9F	0012A	PUSHAB	P.APP	3795
			55	DD	0012D	PUSHL	R5		
		66	02	FB	0012F	CALLS	#2, STR\$COMPARE		
			50	D5	00132	TSTL	R0		
			19	13	00134	BEQL	9\$		
			7C	A8	9F	00136	PUSHAB	P.APR	3796
			55	DD	00139	PUSHL	R5		
		66	02	FB	0013B	CALLS	#2, STR\$COMPARE		
			50	D5	0013E	TSTL	R0		
			0D	13	00140	BEQL	9\$		
			0088	C8	9F	00142	PUSHAB	P.APT	3797
			55	DD	00146	PUSHL	R5		
		66	02	FB	00148	CALLS	#2, STR\$COMPARE		
			50	D5	0014B	TSTL	R0		
			05	12	0014D	BNEQ	10\$		
			80	8F	88	0014F	9\$: BISB2	#128, QUALIFIERS+3	3800
			08	AE	9F	00154	10\$: PUSHAB	RESULT	3803
			010C	C5	9F	00157	PUSHAB	SUMMARY_KEYWORD	
			55	DD	0015B	PUSHL	R5		
		6A	03	FB	0015D	CALLS	#3, LIB\$LOOKUP_KEY		
		74	50	E9	00160	BLBC	STATUS, 14\$		
		52	08	AE	DD	00163	MOVL	RESULT, R2	3804
0C		A443	62	DD	00167	MOVL	(R2), SUM_KEY_VALUE[1]		
			04	A2	DD	0016C	PUSHL	4(R2)	3805

	04	A4	9F	0016F	PUSHAB	SUM_KEY FAO	
67		02	FB	00172	CALLS	#2, STR\$APPEND	
3F		50	E9	00175	BLBC	STATUS, 12\$	
	08	A2	DD	00178	PUSHL	8(R2)	3806
	FF64	C4	9F	0017B	PUSHAB	REPORT HDR1 FAO	
67		02	FB	0017F	CALLS	#2, STR\$APPEND	
32		50	E9	00182	BLBC	STATUS, 12\$	
	08	A2	DD	00185	PUSHL	8(R2)	3807
	FF74	C4	9F	00188	PUSHAB	REPORT HDR3 FAO	
67		02	FB	0018C	CALLS	#2, STR\$APPEND	
79		50	E9	0018F	BLBC	STATUS, 16\$	
		53	D6	00192	INCL	I	3778
		FF29	31	00194	BRW	5\$	
6E	FF64	C4	3C	00197	MOVZWL	REPORT_HDR1_FAO, (SP)	3815
		5E	DD	0019C	PUSHL	SP	
	FF6C	C4	9F	0019E	PUSHAB	REPORT HDR2 FAO	
00000000G	00	02	FB	001A2	CALLS	#2, STR\$DUPE_CHAR	
79		50	E9	001A9	BLBC	STATUS, 17\$	
	0094	C8	9F	001AC	PUSHAB	P.APV	3830
	FF7C	C4	9F	001B0	PUSHAB	REPORT DET FAO	
67		02	FB	001B4	CALLS	#2, STR\$APPEND	
7F		50	E9	001B7	BLBC	STATUS, 20\$	
		53	D4	001BA	CLRL	I	3832
		55	DD	001BC	PUSHL	R5	3833
	00A4	C8	9F	001BE	PUSHAB	P.APX	
69		02	FB	001C2	CALLS	#2, CLISGET_VALUE	
67		50	E9	001C5	BLBC	R0, 18\$	
	08	AE	9F	001C8	PUSHAB	RESULT	3835
	0190	C5	9F	001CB	PUSHAB	REPORT_KEYWORD	
		55	DD	001CF	PUSHL	R5	
6A		03	FB	001D1	CALLS	#3, LIB\$LOOKUP_KEY	
19		50	E8	001D4	BLBS	STATUS, 15\$	
		55	DD	001D7	PUSHL	R5	
		01	DD	001D9	PUSHL	#1	
	009F10FC	8F	DD	001DB	PUSHL	#10424572	
00000000G	00	03	FB	001E1	CALLS	#3, LIB\$SIGNAL	
	50	009F10FC	8F	001E8	MOVL	#10424572, R0	
			04	001EF	RET		
52	08	AE	D0	001F0	MOVL	RESULT, R2	3836
84 A443		52	D0	001F4	MOVL	R2, REPORT_VALUE[1]	
50	08	A2	D0	001F9	MOVL	8(R2), R0	3837
74		9440	DE	001FD	MOVAL	@BUCKET_SIZE[R0], BUCKET_SIZE	
	0C	A2	DD	00201	PUSHL	12(R2)	3838
	FF7C	C4	9F	00204	PUSHAB	REPORT DET FAO	
67		02	FB	00208	CALLS	#2, STR\$APPEND	
28		50	E9	0020B	BLBC	STATUS, 20\$	
	10	A2	DD	0020E	PUSHL	16(R2)	3839
	FF64	C4	9F	00211	PUSHAB	REPORT HDR1 FAO	
67		02	FB	00215	CALLS	#2, STR\$APPEND	
1E		50	E9	00218	BLBC	STATUS, 20\$	
	14	A2	DD	0021B	PUSHL	20(R2)	3840
	FF6C	C4	9F	0021E	PUSHAB	REPORT HDR2 FAO	
67		02	FB	00222	CALLS	#2, STR\$APPEND	
11		50	E9	00225	BLBC	STATUS, 20\$	
	FC	A4	D6	00228	INCL	REPORT_ITEMS	3841
		53	D6	0022B	INCL	I	3833
		8D	11	0022D	BRB	13\$	

ACC
V04-000

I 12
15-Sep-1984 23:36:11 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 11:51:58 DISK\$VMSMASTER:[ACC.SRC]ACC.B32;1

Page 114
(16)

F
V

64	D5	0022F	18\$:	TSTL	BUCKET_SIZE	:	3849
03	13	00231		BEQL	19\$	:	
04	C0	00233		ADDL2	#4, BUCKET_SIZE	:	
01	D0	00236	19\$:	MOVL	#1, R0	:	3851
04	00239	20\$:	RET			:	3853

; Routine Size: 570 bytes, Routine Base: CODE + 2664

ACC
V04-000

J 12
15-Sep-1984 23:36:11
14-Sep 1984 11:51:58

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[ACC.SRC]ACC.B32;1

Page 115
(17)

: 3032 3854 1 END
: 3033 3855 0 ELUDOM

.EXTRN LIB\$SIGNAL

PSECT SUMMARY

Name	Bytes	Attributes
DATA	2280	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
LIB\$KEYOS	0	NOVEC, NOWRT, RD, EXE, SHR, LCL, REL, CON, PIC, ALIGN(1)
LIB\$STATES	10	NOVEC, NOWRT, RD, EXE, SHR, LCL, REL, CON, PIC, ALIGN(1)
CODE	10398	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$OWNS	788	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$GLOBALS	384	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	168	1	581	00:01.0
\$255\$DUA28:[SYSLB]TPAMAC.L32;1	42	19	45	14	00:00.2

COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD,'INITIAL,OPT,SIZE)/LIS=LISS:ACC/OBJ=OBJ\$:ACC MSRC\$:ACC/UPDATE=(ENHS:ACC)

: Size: 6207 code + 7653 data bytes
: Run Time: 02:44.1
: Elapsed Time: 04:10.8
: Lines/CPU Min: 1409
: Lexemes/CPU-Min: 29782
: Memory Used: 767 pages
: Compilation Complete

0001 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY